

# 任意位置クエリに適用可能な効率的逆最遠傍検索

劉 健全<sup>†</sup> 陳 漢雄<sup>†</sup> 古瀬 一隆<sup>†</sup> 北川 博之<sup>†,††</sup>

<sup>†</sup> 筑波大学大学院システム情報工学研究科 〒305-8577 茨城県つくば市天王台 1-1-1

<sup>††</sup> 筑波大学計算科学研究センター 〒305-8577 茨城県つくば市天王台 1-1-1

E-mail: †ljq@dblab.is.tsukuba.ac.jp, ††{chx, furuse, kitagawa}@cs.tsukuba.ac.jp

あらまし 本論文では、逆最遠傍 (Reverse Furthest Neighbors, RFN) 検索という新たな問合せ問題に対して、与えられた任意位置のクエリ  $q$  に対応できる効率的な検索手法を提案する。先行研究では、従来手法の高い計算量を大幅に削減できたが、任意位置、特にデータセットに遠く離れたクエリに有効的ではない。本研究では、位置に依存しないクエリ  $q$  の特別な性質を発見し、効率的な検索アルゴリズムを提案する。さらに、合成データと実データを用いて提案手法の効率とスケーラビリティを検証する。

キーワード 逆最遠傍 (RFN), 任意位置, 問合せ処理, 距離索引, 最近傍検索

## 1. はじめに

過去十年間、類似検索の科学研究・開発への応用が急増している。例として、パターン認識や [1], [2], 画像検索 [3], 時系列マッチング [4] などが挙げられる。これらの応用に伴って、類似検索に関する研究は活発にされている。類似検索に関する問い合わせ処理は、 $k$  近傍 ( $k$ -Nearest Neighbors,  $k$ -NN) や、範囲問い合わせ (Range Query), 逆最近傍 (Reverse Nearest Neighbors) などの変形である。これらに対するオンライン問い合わせ処理の要求が急増しており、様々な処理技術が提案された。それらは主に、空間分割法 (例: grid-file [5]), データ索引法 (例: R-tree [6]), 連続スキャン法 (例: VA-file [7]) との三種類に分けられる。

しかし、近傍の反対、すなわち、遠傍に対する問い合わせ処理はこれまで今後の課題とされ、特に逆最遠傍 (Reverse Furthest Neighbors, RFN) の問い合わせ処理は未着手となっていた。しかし、実世界の応用では、効率的な逆最遠傍の問い合わせ処理は有意義である。Yao たち [8] は、初めて逆最遠傍問題に関する正式な定義を示し、R-tree に基づく問い合わせ処理方法とともに提案した。あるデータセット  $O$  と任意の

クエリ  $q$  が与えられたとき、 $O$  の中から  $q$  を最遠傍とする全てのオブジェクトの集合を、 $q$  の逆最遠傍 (RFN) と言う。また、この問題を解くことを逆最遠傍検索と言う。

例として、図 1 を用いて説明する。図 1 では、RFN の問い合わせ処理を示している。オブジェクト集合  $O$  及びクエリ  $q$  を与えられたとき、 $q$  の逆最遠傍は集合  $\{o_3, o_4\}$  になる。単純な方法でこれを求めるには、各オブジェクト  $o_i$  において、 $o_i$  と  $q$  の距離、及び  $o_i$  と他の  $o_j (i \neq j)$  の距離を算出し、 $o_i$  は  $q$  を最遠傍とすることが確認すればよい。 $o_3$  と  $o_4$  のみが  $q$  をそれぞれの最遠傍 (それぞれの距離は破線で表す) とするので、 $q$  の逆最遠傍の解になることが分かる。

次に、逆最遠傍問題を求めるのが有意義であることを示すため、我々の先行研究 [9] に基づき、以下に具体的な応用例を示す。

応用例 1: 逆最遠傍問題は観光産業において応用することができる。例えば、東京都にある全ての観光地をデータセット  $O$  とする。観光地のあたりにある様々なお店に対して、普段、観光客はその観光地に近い店を利用する。特別な理由がなければ、より遠くの店へは行かない。店の経営者の立場からすれば、遠くてもより多くの顧客に来店してほしいと考える。したがって、経営者にとっては、自分の店を最遠傍 (あるいは観光客が最も行きたくない) とする観光地において、店の宣伝の力を入れるべきである。この場合、宣伝をするべき観光地はこの店 ( $q$ ) の逆最遠傍となる。

応用例 2: 都市計画の分野でも応用例が挙げられる。例えば、自治体の都市計画部門では、建築申請を審査する際、市民への影響や自治体の計画などを含めている検討する。例えば、ある化学工場を建築する提案がなされたときに、

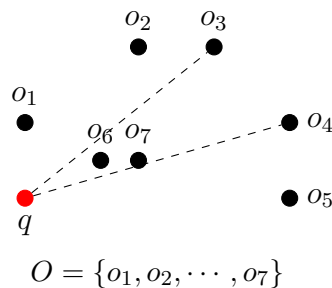


図 1 RFN の問い合わせ例。

自治体が一番考慮するのは、その化学工場の建築候補地は住民にどのくらい影響するかということである。もちろん、影響しないあるいは影響の最も少ない候補地のほうが建築する場所として好ましい。このような候補地の選択は、逆最遠傍問題によって行うことができる。この例の場合、建設候補地と住宅との距離が遠ければ遠いほど、住民への影響が少ないと言える。化学工場の建築候補地集合を  $Q$  とし、住宅の所在地集合を  $O$  とすると、候補地  $q \in Q$  の逆最遠傍になる  $o \in O$  の数が一番多い  $q$  は最適な候補地となる。

本論文では、逆最遠傍問題に対して、与えられた任意位置のクエリ  $q$  に対応でき、先行研究 [9] で提案された方法より効率的な検索手法を提案する。本論文の構成は以下の通りである。第 2 章で関連研究及び先行研究を紹介する。第 3 章では、本研究の提案手法の基本定理を詳しく説明する。続いて、第 4 章で本研究のアルゴリズムを提案し、第 5 章で評価実験によって本提案手法が効率向上であることとスケーラビリティを示す。最後に、本論文での内容を簡単にまとめ、今後の課題を述べる。

## 2. 関連研究

類似検索に関する技術はこれまでもよく研究されていた。本研究と最も関連のある問い合わせタイプは主に 3 つのタイプに分けられる。すなわち、(k-) 最近傍検索 (NN, あるいは k-NN)、逆 (k-) 最近傍検索 (RNN, あるいは RkNN)、逆最遠傍 (RFN) である。

最近傍検索 については、早期に R-tree [6] を用いる深さ優先探索法 [10], [11] と最良優先探索法 [12] が提案された。近年、Athitsos たちは距離に基づくハッシュ手法を用いた近似最近傍検索を提案した [13]。また、Tao たちは LSB-tree による高次元データにおける最近傍検索を提案した [14]。

逆最近傍検索 に関しては、Korn と Muthukrishnan が初めて逆最近傍の問い合わせを提案した [15]。それ以降、様々な問い合わせ処理手法が提案されておる。例えば、RdNN-Tree [16], TPL [17], MRkNNCop-Tree [18] などがある。これらには、大量なデータに対する RNN 応用例も示されている。

逆最遠傍検索 について、本研究と最も関連するものは [8] である。Yao たちは初めて逆最遠傍検索 RFN を定義した。RFN の問い合わせを処理するため、彼らは Progressive Furthest Cell (PFC) と Convex Hull Furthest Cell (CHFC) とのアルゴリズムを提案した。これらのアルゴリズムでは、R-tree とともに、最遠ボロノイセル (furthest Voronoi cell, fvc) を利用して、あるデータオブジェクト  $o \in O$  が  $q$  の RFN になるかどうかを判断する。ここで、 $fvc(q, O)$  は、データセット  $O$  における、クエリ  $q$  に関する凸多辺形を表す。 $fvc(q, O)$  を計算するには、 $q$  と  $o (o \in O)$  の全てのペアの垂直二等分線を引いて、データ空間を 2 つ部分空間に分けて、 $q$  と離れ

ている部分空間をそれぞれの交差を取ったものがクエリ  $q$  に関する凸多辺形領域になる。この領域が  $q$  の逆最遠傍になる  $o \in O$  を厳密に囲むことが証明されている。この性質を利用するため、PFC アルゴリズムでは、R-tree における走査をしながら、 $fvc(q, O)$  を構築する。同時に、 $fvc(q, O)$  を用いて、各 MBR あるいはデータオブジェクトが解になるかをチェックする。しかし、このアルゴリズムでは、解にならないものはそのまま捨てられない。解にならないものを用いて、より狭い  $fvc(q, O)$  領域を再構築することは可能であるため、この後処理は非常に非効率的である。PFC を改善するため、彼らはさらに CHFC アルゴリズムを提案した。CHFC では、逆最遠傍問題に対する特有な凸包特徴を示し、この特徴をさらに利用して  $fvc(q, O)$  を構築することで、より効率のよい検索を行う。CHFC アルゴリズムは PFC より速いが、効率の悪い処理も実際には含まれている。例えば、凸包を計算すること、R-tree を用いた範囲問い合わせを実行することは計算コストが高い。

これらの高価な計算コストを削減するため、先行研究では新たなアプローチを提案した。先行研究において効率向上のための目標は、1) 凸包の計算を避けたること、2) R-tree を使った範囲問い合わせの計算を避けること、3) 検索するとき高価な距離計算をなるべく削減すること、であった。先行研究で提案したアルゴリズムは Yao たちの従来手法より計算の高速化が実現できたが、任意位置、特にデータセットに遠く離れたクエリに有効ではないという問題がある。この場合にも、先行研究の手法はそのままフィルタリングをかけて検索を行う。しかし、この際にフィルタリングが不要であり、データセット内の全てのオブジェクトが逆最遠傍の解になるような領域が存在する。よって、より完全なフィルタリング条件を発見するのは必要になる。このため、本研究では、位置に依存しないクエリの特別な性質を発見し、フィルタリング条件の理論定義及び証明を行い、効率を向上させるアルゴリズムを提案する。

## 3. 任意位置に適用したフィルタリング

本章では、第 2 章に述べたチャレンジに向け、本研究のアプローチの理論的な根拠となる概念と結論を詳しく説明する。本稿は先行研究の改良を行うため、先行研究で提案した定理と補題、及びピボット (凸包) の構成と距離索引の構造を利用する。詳しくはもとの論文 [9] にて確認ができる。本稿では、クエリ  $q$  の位置を任意にすると考えられる。問い合わせの処理を行うときに、この「任意位置」を 2 つの場合に分けて討論する。1) データセットとの近い範囲にクエリ  $q$  が与えられたときに、先行研究の手法をそのまま利用する。2) データセットに遠く離れたクエリ  $q$  が与えられたときに、本稿で定義する「安全区域」とフィルタリングの条件及びアルゴリズムを使う。よって、本章では、 $q$  がデータセットに遠

く離れている場合のみに対して、本研究で特有する定義、定理と証明を述べる。

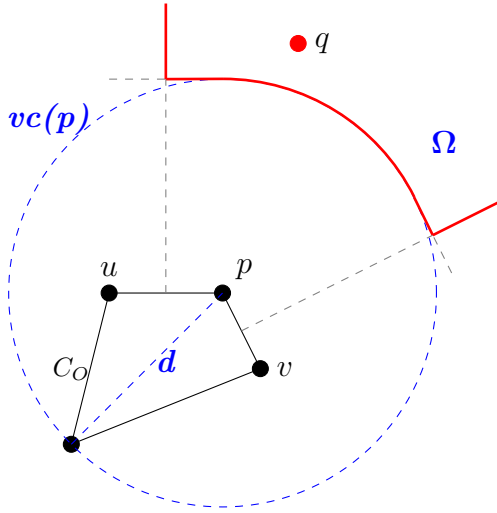


図2 凸包による定義：直径  $d$ ，頂点円  $vc(p)$ ，安全区域  $\Omega$

まず、図2で表すようにデータセットに遠く離れたクエリ  $q$  に適用するため、凸包の直径、頂点円、安全区域についての定義を説明する。

[定義1] (凸包の直径) データセット  $O$  及びその凸包  $C_O$  に関して、凸包の任意2つ異なる頂点の最大距離を凸包の直径という。凸包の直径を記号  $d$  を使い、 $d = \max\{dist(u, v) | u \neq v \wedge u, v \in C_O\}$  で表す。

図2では、指定したデータセットとその凸包  $C_O$  に関して、破線で引いて  $d$  で表しているのは凸包  $C_O$  の直径である。この定義によって、 $d$  はデータセットの中に任意の2つのオブジェクトの距離より大きい、距離の上界になることが分かる。すなわち、[19]の証明により、 $d$  はデータセットの直径であることも成り立つ。直径  $d$  を定義するには、逆最遠傍問題を解くときに、あるクエリ  $q$  がデータセットにおいて  $\forall o, |oq| > d$  が成り立てば、データセットの全体は  $q$  の逆最遠傍の解になる。直径  $d$  に関する特徴を利用することで、条件  $|oq| > d(\forall o)$  を保証できる安全区域を構築するために必要な頂点円の定義を次に説明する。

[定義2] (頂点円) 与えられた凸包  $C_O$  に対して、凸包の任意1つの頂点  $p (\forall p \in C_O)$  を中心として、 $d$  を半径として描いた円は頂点円という。頂点  $p$  に関する頂点円を記号  $vc(p)$  で表す。

図2では、頂点  $p$  を円心として、 $d$  を半径として描いた頂点円  $vc(p)$  の例を示している。

[定義3] (安全区域) データセット  $O$  及びその凸包  $C_O$  と、任意位置のクエリ  $q$  が与えられたときに、次の手順で作成した区域を安全区域という。

- 凸包  $C_O$  に対して、クエリ  $q$  の最近傍になる頂点  $p$  を探す。

- 凸包  $C_O$  上に頂点  $p$  と隣接する2つの頂点  $u, v$  を探す。
- 頂点  $p$  に関する頂点円  $vc(p)$  を描く。
- 線分  $\overline{up}, \overline{vp}$  とそれぞれ平行する頂点円  $vc(p)$  の接線を引く。
- 線分  $\overline{up}, \overline{vp}$  に対して、それぞれの垂直二等分線を引く。接線と二等分線で分割した  $q$  に向け空間、 $vc(p)$  の外部である交差区域を安全区域  $\Omega$  という。

具体例を挙げると、図2では、赤線で引いて  $\Omega$  で表す外側の領域が安全区域を示している。この安全区域の定義により、我々は任意位置のクエリ  $q$  に適応できる逆最遠傍判定条件を発見した。下記の定理を記述する。

[定理1] 与えられたデータセット  $O$  に対して、その凸包  $C_O$  の外側にある任意位置のクエリ  $q$  について、 $q$  が安全区域  $\Omega$  に属するとき、データセット  $O$  の全てのオブジェクトは  $q$  の逆最遠傍の解になる。

[証明1] 定理1を証明するため、クエリ  $q$  の位置によって、次の2つ場合に分けて議論する。

- (1) クエリ  $q$  が安全区域  $\Omega$  の弧の垂直方向にあるとき。
- (2) クエリ  $q$  が安全区域  $\Omega$  の接線の垂直方向にあるとき。

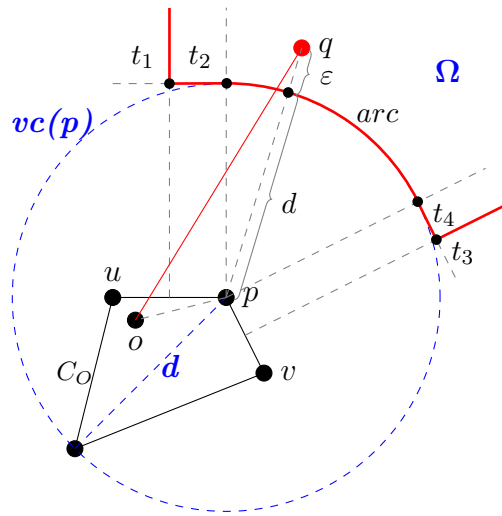


図3 定理の証明：場合1；弧  $\widehat{t_2t_4}$  の垂直方向

[場合1] 図3にデータセット中の任意のオブジェクト  $o$  に対して、クエリ  $q$  が安全区域  $\Omega$  の弧  $\widehat{t_2t_4}$  の垂直方向にあるときに、下記の前提は成り立つ。

- 線分  $\overline{t_1t_2} \parallel \overline{up}$  (平行) ,
- 線分  $\overline{pt_2}$  が頂点円  $vc(p)$  の半径である ( $r = d$ ) .
- かつ、線分  $\overline{t_1t_2}$  が頂点円  $vc(p)$  の接線である。
- $\Rightarrow \overline{t_1t_2} \perp \overline{pt_2}$ , 角  $\angle pt_2t_1 = \angle upt_2 = 90^\circ$  となる。

次に、三角形  $\triangle opq$  について、角  $\angle opq \geq 90^\circ$  であるため、次のことは成り立つ。

$$\angle opq + \angle pqo + \angle qop = 180^\circ$$

$$\Rightarrow \angle opq \geq 90^\circ \geq \angle pqo + \angle qop > \angle qop$$

三角形の正弦定理により

$$\Rightarrow |\overline{oq}| > |\overline{pq}| = d + \varepsilon > d \ (\varepsilon > 0)$$

したがって、場合 1 では、データセット内部の任意のオブジェクト  $o$  と  $q$  の距離  $|\overline{oq}|$  は凸包の直径  $d$  より大きい。 $o$  と  $o$  以外のどのオブジェクトとの距離と比べても、当然に  $\text{dist}(o, q)$  は最大になる、すなわち、 $q$  は  $o$  の最遠傍になる。 $o$  はデータセット中の任意オブジェクトなので、逆最遠傍の定義によって、この場合にデータセットの全体は  $q$  の逆最遠傍の解になる。

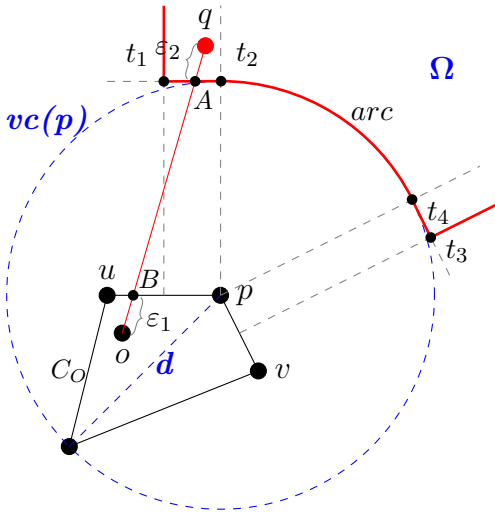


図 4 定理の証明：場合 2； 接線  $\overline{t_1t_2}$  または  $\overline{t_3t_4}$  の垂直方向

また、図 4 のように、クエリ  $q$  が安全区域  $\Omega$  の接線  $\overline{t_1t_2}$  または  $\overline{t_3t_4}$  の垂直方向にあるときに、頂点円の圆心  $p$  によって、 $p$  に隣接する凸包の頂点  $u, v$  の 2 つある。それぞれに対応する接線の方向が異なるが、 $q$  に関する性質は変わらないので、証明を簡単にするため、左側に隣接する頂点  $u$  と  $p$  の組のみ証明する。

[場合 2] データセット中の任意オブジェクト  $o$  に対して、クエリ  $q$  が安全区域  $\Omega$  の接線の垂直方向にあるときに、下記のことが成り立つ。

$$\because \text{線分 } \overline{t_1t_2} \parallel \overline{up} \perp \overline{pt_2}$$

$A$  と  $B$  は線分  $\overline{oq}$  と平行線との交点とすれば、

$d = \overline{pt_2}$  が平行線間の垂直距離である。

$$\Rightarrow \overline{AB} \geq d$$

$$\because \varepsilon_1 \geq 0, \varepsilon_2 > 0$$

$$\Rightarrow \overline{oq} = \overline{oB} + \overline{AB} + \overline{Aq} = \varepsilon_1 + d + \varepsilon_2 > d$$

したがって、場合 1 と同様に、場合 2 では、データセット

の全体が  $q$  の逆最遠傍の解になる。また、特別のケース、 $q$  は場合 1 と 2 の境界線上にあるとき、例えば、 $q$  が  $pt_2$  または  $pt_4$  上に頂点円  $vc(p)$  の外側にあるときに、場合 1, 2 のどちらにも適用可能であり、定理 1 の結論にも成り立つ。

以上の証明により、安全区域  $\Omega$  に入ってくる  $q$  にとって、データセット全体が  $q$  の逆最遠傍の解になるのが分かる。

#### 4. アルゴリズム

第 3 章で述べた「安全区域」に関する結論に基づいて、本章では任意位置のクエリ  $q$  に適用できるかつ効率的なアルゴリズムを提案する。先行研究 [9] の距離索引と PIV アルゴリズムの改良版として、PIV<sup>+</sup> と呼ぶ。

PIV<sup>+</sup> アルゴリズムでは、まず、与えられた任意位置のクエリ  $q$  に対して、入力した凸包 (ピボット)  $S_{piv}$  を使い、定義 3 によって、安全区域  $\Omega$  を求める (1 行目)。次に、定理 1 を条件と扱い、クエリ  $q$  が安全区域  $\Omega$  に入っているかどうかを判断する (2 行目)。入ってあれば、データセットの全体を  $q$  の逆最遠傍の解として問い合わせ処理を終了する (3 行目)。これ以外の場合 (4 行目) は、引数をそのまま先行研究の PIV アルゴリズムに渡して、PIV の計算結果を戻す (5 行目)。

任意位置のクエリ  $q$  を与えるとき、先行研究の PIV アルゴリズムでは、ピボットと距離索引を利用していたフィルタリングが、データセットに遠く離れたクエリに有効的ではないので、フィルタリングの効率が著しく低下する。一方、PIV<sup>+</sup> アルゴリズムでは、定理を利用することにより、フィルタリングのための無駄な計算を有効に避けることが分かる。

**Algorithm 1:** PIV<sup>+</sup>(Query  $q$ , Pivots  $S_{piv}$ , Dataset  $O$ )

**Require:**  $q$ , query object;  $O$ , dataset;  $S_{piv}$ , convex hull of  $O$ .

- 1: Compute safe area  $\Omega$ ; { Definition 3 }
- 2: if  $q \in \Omega$  then
- 3:   return  $O$ ; { Theorem 1 }
- 4: else
- 5:   return PIV( $q, S_{piv}, O$ ); { Previous algorithm [9] }
- 6: end if

#### 5. 実験と評価

本論文の提案手法の有効性を確認するため、実験を行った。実験環境は以下の通りである。

- CPU: Intel(R) Xeon (R) 2.83 GHz
- 主記憶: 16 GB
- 実装言語: C++

実験データには、合成データと実データを用いた。合成データは一様分布 (UN) とランダムクラスタ分布 (RC) との 2 種類のデータを使った。実データはアメリカの道路ネット

ワーク図 (Map) を利用した。(注1)

上記の環境で合成データと実データのそれぞれについて、先行研究 [9] の PIV 法と、関連研究 [8] の計算方法 CHFC と、力まかせ探索 BFS と、本研究のアルゴリズム PIV<sup>+</sup> との比較実験を行った。計算コストを比較項目として評価を行った。以下にその結果を示す。

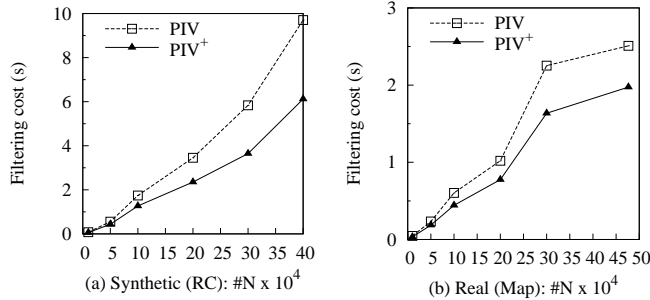


図5 フィルタリングのコスト：(a) 合成データ，(b) 実データ。

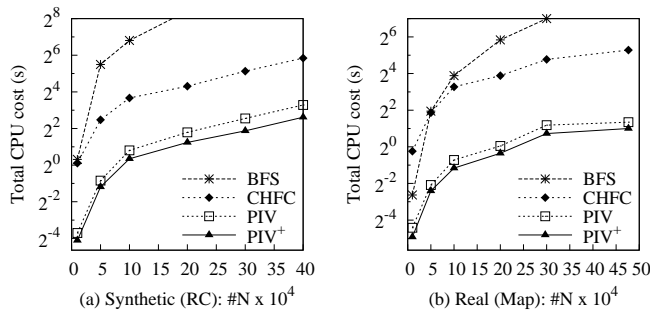


図6 アルゴリズム全体コスト：(a) 合成データ，(b) 実データ。

まず、図5では、合成データ (RC) および実データ (Map) において、先行研究の PIV 法と本論文の PIV<sup>+</sup> 法とのフィルタリング段階での計算コストの比較を示す。ここで、PIV<sup>+</sup> 法は PIV 法よりフィルタリングの計算効率を2倍ほど速くなることが分かった。スケーラビリティとえば、データセットに関わらずにデータオブジェクトの数 (N) が増えるほど、フィルタリングにかかる計算時間は線形に増加することが見られた。

また、BFS 法、CHFC 法及び PIV 法とのアルゴリズムの全体計算コストを比較して、図6より本研究の提案手法の効率とスケーラビリティを評価した。図6の(a)では、合成データを用い、(b)では実データを用いた。それぞれのデータにおける計算コストを示す。結果によって、逆最遠傍問題に対して、本提案の PIV<sup>+</sup> 法が最も速い問い合わせ処理をすることが実証された。CHFC 法に比べると、処理速度で約10倍の向上が見られた。先行研究の PIV よりも、処理速度は約2倍で向上させた。さらに、データセット O のサイズの

変化により、CHFC 法と BFS 法より、PIV 法と PIV<sup>+</sup> 法のほうがより安定的であることが分かった。

## 6. まとめ

本論文では、逆最遠傍 (RFN) という新たな問合せ問題に対して、与えられた任意位置のクエリ  $q$  に対応できる効率的な検索手法を提案した。本稿の基本定義・定理を導入したことで、任意位置、特にデータセットに遠く離れたクエリに対して、先行研究で提案された PIV 手法が有効ではない問題を解決するとともに、効率をさらに向上させる検索アルゴリズム PIV<sup>+</sup> を示した。また、合成データと実データを用いて提案手法の効率とスケーラビリティの検証から、先行研究の手法と従来手法より顕著に効果があることを示した。

今後の課題として、本提案手法をさらに汎用的なものとするため、I/O コストの考慮も含めた手法の改良や拡張などを行うとともに、より詳細な評価実験の実施を行うことを検討している。

謝辞 本研究の一部は科学研究費補助金特定領域研究 (21013004) による。

## 文献

- [1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
- [2] Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover’s distance. In *CVPR (1)*, pages 220–227, 2004.
- [3] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Ze Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2), 2008.
- [4] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient similarity search in sequence databases. In *FODO*, pages 69–84, 1993.
- [5] Jürg Nievergelt, Hans Hinterberger, and Kenneth C. Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Trans. Database Syst.*, 9(1):38–71, 1984.
- [6] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD Conference*, pages 47–57, 1984.
- [7] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, pages 194–205, 1998.
- [8] Bin Yao, Feifei Li, and Piyush Kumar. Reverse furthest neighbors in spatial databases. In *ICDE*, pages 664–675, 2009.
- [9] Jianquan Liu, Hanxiong Chen, Kazutaka Furuse, and Hiroyuki Kitagawa. An efficient algorithm for reverse furthest neighbors query with metric index. In *DEXA*, pages 437–451, 2010.
- [10] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest neighbor queries. In *SIGMOD Conference*, pages 71–79, 1995.
- [11] King Lum Cheung and Ada Wai-Chee Fu. Enhanced nearest neighbour search on the r-tree. *SIGMOD Record*, 27(3):16–21, 1998.
- [12] Gísli R. Hjaltason and Hanan Samet. Distance brows-

(注1): <http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm>

- ing in spatial databases. *ACM Trans. Database Syst.*, 24(2):265–318, 1999.
- [13] Vassilis Athitsos, Michalis Potamias, Panagiotis Papapetrou, and George Kollios. Nearest neighbor retrieval using distance-based hashing. In *ICDE*, pages 327–336, 2008.
  - [14] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. Quality and efficiency in high dimensional nearest neighbor search. In *SIGMOD Conference*, pages 563–576, 2009.
  - [15] Flip Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *SIGMOD Conference*, pages 201–212, 2000.
  - [16] Congjun Yang and King-Ip Lin. An index structure for efficient reverse nearest neighbor queries. In *ICDE*, pages 485–492, 2001.
  - [17] Yufei Tao, Dimitris Papadias, and Xiang Lian. Reverse knn search in arbitrary dimensionality. In *VLDB*, pages 744–755, 2004.
  - [18] Elke Aichtert, Christian Böhm, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. Efficient reverse k-nearest neighbor search in arbitrary metric spaces. In *SIGMOD Conference*, pages 515–526, 2006.
  - [19] Binay K. Bhattacharya and Godfried T. Toussaint. Fast algorithms for computing the diameter of a finite planar set. *The Visual Computer*, 3(6):379–388, 1988.