

空間データベースシステムのための可視探索手法

能登谷淳一[†] 杉本 雄太^{††} 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 秋田県由利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 秋田県由利本荘市土谷字海老ノ口 84-4
E-mail: †{notoya,m07b007,kusakari,kasai}@akita-pu.ac.jp

あらまし 与えられた点から可視である空間オブジェクトを発見する問題は、主にコンピュータグラフィックスの分野で着目されてきた問題である。近年、空間データベースシステムの応用分野拡大に伴い、多量の空間オブジェクトに対して可視問題など各種の幾何問題を高速に解決することが求められている。本研究では、空間データベースに格納された空間オブジェクト中から可視である可能性のあるオブジェクトの集合を高速に発見するフィルタリング手法を提案する。提案アルゴリズムは空間走査法のアイデアを応用し、空間オブジェクトが格納された木構造索引中を効率的に走査するアルゴリズムである。実験的評価の結果、提案手法を用いた可視オブジェクトフィルタリングによる可視探索時の入出力コストの大幅な削減が確認された。

キーワード 可視判定, 遮蔽物抽出, 平面走査法, 木構造空間索引

Visibility Search for Spatial Database Systems

Junichi NOTOYA[†], Yuta SUGIMOTO^{††}, Yoshiyuki KUSAKARI[†], and Masao KASAI[†]

[†] Faculty of Systems Science and Technology, Akita Prefectural University
Ebinokuchi 84-4, Tsuchiya, Yuri-Honjyo, Akita 015-0055 Japan

^{††} Graduate School of Systems Science and Technology, Akita Prefectural University
Ebinokuchi 84-4, Tsuchiya, Yuri-Honjyo, Akita 015-0055 Japan

E-mail: †{notoya,m07b007,kusakari,kasai}@akita-pu.ac.jp

Abstract To find spatial objects that are visible from a given point is an important problem in computer graphics. Recently, many applications of spatial database systems need to solve various geometric problems such as visibility searching. In this paper, we present an efficient algorithm to find the candidates of visible objects which are originally stored in the spatial database systems. Our space sweep algorithm traverses on spatial index tree efficiently. We also present the results of experiments on the implementation of our filtering algorithm. It shows that our algorithm reduces total I/O cost significantly while visibility search.

Key words Visibility determination, occlusion culling, plane sweep algorithm, spatial index trees

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。特に空間データベースシステムにおいては、格納された多量の空間オブジェクトに対して各種の幾何探索問題を高速に解決する索引機構の提供が求められている [1]。

可視探索問題は広い応用範囲を持つ幾何探索問題の一

つであり、空間オブジェクト集合と問い合わせ図形が与えられたとき、問い合わせ図形から可視であるような空間オブジェクトを抽出する問題である。可視探索問題については過去にコンピュータグラフィクス分野で多くの研究が行われてきた。しかし、それらの研究の大部分はコンピュータグラフィクス分野の特性を利用したものであった。これに対し、本研究ではより広い分野への可視探索の応用を想定し、空間データベース中の一般の大規模空間オブジェクト集合に対する可視探索アルゴリズムを提案する。

本稿で提案する探索アルゴリズムは、R木[2]などの木構造空間索引[1]を利用して索引付けされた空間オブジェクト集合に対する可視探索アルゴリズムである。提案アルゴリズムは木構造空間索引を利用したk近傍探索アルゴリズムで利用されている空間オブジェクト間の距離尺度[3],[4]と同様の尺度を利用した空間走査手法により、任意の形状を持つオブジェクト集合に対する可視探索を高速に実現する。

本研究で提案する探索アルゴリズムを利用することにより、空間データベース全体を参照することなく、データベース中の全ての可視オブジェクトを抽出することが可能となる。

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、可視探索に関わるいくつかの性質について述べる。第4節では木構造空間索引上の空間走査法を利用した可視探索手法の提案を行う。第5節では本研究で提案する可視探索手法の実験的性能評価の結果を示す。第6節は本稿のむすびである。

2. 関連研究

可視探索問題は、従来コンピュータグラフィクスの分野で着目され、研究が行われてきた幾何探索問題である。コンピュータグラフィクスの分野では、可視探索問題を主に三次元レイトレーシングの前段処理と捉えた研究が行われてきた。

三次元レイトレーシングの主たる目的は、表示画面上の各画素における画素値の計算である。そのため、コンピュータグラフィクス分野における可視探索問題の研究では、各画素の中心を通る半直線上での可視探索が主要な研究テーマであった。このような応用においては、ある一点から可視である空間オブジェクトの集合は各画素を通る半直線上の可視オブジェクト集合の和集合によって近似される。各画素に対応する可視オブジェクトの発

見には、zバッファ法やその発展アルゴリズム[5]、BSP木[6]を利用した可視探索の高速化[7]などが提案されている。

可視探索問題に類似の問題として、空間オブジェクト間の遮蔽関係判定問題がある。可視探索問題を遮蔽関係判定により解決するための手法として、探索の対象となる空間オブジェクトの集合を遮蔽物集合(occluders)と被遮蔽物集合(occludees)の2個の空間オブジェクト集合にあらかじめ分割し、特定の遮蔽物に遮蔽される被遮蔽物を探索の対象から除外することで可視探索の探索空間を縮小する手法[8],[9]が提案されている。

これらのコンピュータグラフィクスの分野における研究の多くは、以下のような応用分野特有の性質のうちのいくつかを利用している。

- 画素集合の画素値を出力とする
 - 近似解を高速に求めることが要求される
 - 画像処理ハードウェアによる支援を利用可能である
 - データがあらかじめ主記憶に格納可能なサイズに縮小されている
 - 探索対象となる空間オブジェクトの更新が頻繁ではない
 - 探索空間の次元数が3である
 - 空間オブジェクトが三角形など特定の形状を持つ
- これに対し、本研究では空間データベースシステム上の可視探索手法の提案を目指し、以下のような条件下での可視探索を考える。
- 問い合わせ点から可視である全ての空間オブジェクトからなる集合を出力とする
 - 探索対象である空間オブジェクトのデータは可視探索以外の用途にも利用されることを想定した形態で二次記憶中に格納されているものとする
 - データの更新(追加, 削除, 位置の変更)が頻繁に発生する
 - 探索空間の次元数に制限を設けない
 - 空間オブジェクトの形状には制限を設けない

本研究で提案する手法は、遮蔽関係の判定に基づく探索空間の縮小手法の一種である。ただし、従来の探索空間縮小手法と異なり、遮蔽物集合と被遮蔽物集合をあらかじめ与えるのではなく、遮蔽物集合を探索アルゴリズムの進行過程に従って動的に更新する。

3. 空間データベースシステムにおける可視探索問題

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 次元ユークリッド空間 \mathbb{R}^d を想定する。

\mathbb{R}^d 空間中の部分空間 $S \subset \mathbb{R}^d$ を領域と呼ぶ。 \mathbb{R}^d 空間上の一点 x , 二点 x, y を結ぶ開線分 \overline{xy} , 点 x を始点とし点 y を通る半直線 \overrightarrow{xy} , 点 x を中心とし半径を r とする d 次元超球の表面など d より小さい次元を持つ対象も領域の一種であるとする。

空間データベースに格納される各空間オブジェクト P は \mathbb{R}^d 上の領域 $S(P)$ と対応する。 $S(P)$ を空間オブジェクト P の領域と呼ぶ。

3.2 点始点可視探索問題

本研究で着目する可視探索問題は, d 次元ユークリッド空間 \mathbb{R}^d 上の可視探索問題のうち, 与えられた一点からの可視オブジェクト集合を求める問題である。この種の問題は点始点可視探索問題 (VFP: Visibility From Point) と呼ばれる。

[定義 1] (点始点可視探索問題) n 個の空間オブジェクトからなる集合 $\mathbb{P} = \{P_1, \dots, P_n\}$ を考える。このとき, 与えられた点 $q \in \mathbb{R}^d$ から可視である \mathbb{P} の全ての要素からなる集合 $\mathbb{V} \subset \mathbb{P}$ を求める問題を \mathbb{R}^d 上の点始点可視探索問題と呼ぶ。ただし, 空間オブジェクト $P \in \mathbb{P}$ が q から可視であるとは, 点 $p \in S(P)$ が存在し, 開線分 \overline{qp} が $\bigcup_{O \in \mathbb{P}} S(O)$ と共通部分を持たないことを言う。点始点可視探索問題の入力のうち, 集合 \mathbb{P} の要素を探索対象オブジェクト, 点 q を問い合わせ点と呼ぶ。

点始点可視探索問題は空間オブジェクトの領域間の遮蔽関係によって記述可能である。

[定義 2] 点 q および領域 O, P に対し, P 中の任意の点 $p \in P$ に対して, 開線分 \overline{qp} が O と共通部分を持つとき, q から見て O は P を隠すという。

明らかに, 空間オブジェクト P が q から可視であることは領域 $\bigcup_{O \in \mathbb{P}} S(O)$ が q から見て $S(P)$ を隠さないことと等価である。

領域の包含関係と遮蔽関係の間には, 以下に示す関係が存在する。

[補題 1] 点 q および領域 O, P, R に対し, O が q から見て R を隠し, かつ $P \subset R$ ならば O は q から見て P を隠す。

証明: 任意の点 $p \in P$ について $p \in R$ より明らか。□

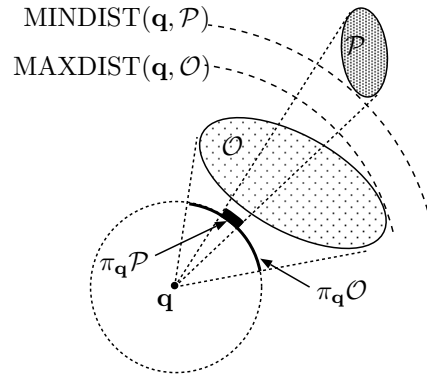


図 1 領域の距離とイメージを利用した遮蔽関係の判定
Fig.1 Occluding determination by distance metrics and projected images of regions.

本研究では, 空間オブジェクト間の遮蔽関係の判定および木構造索引上の走査順序決定のために, \mathbb{R}^d 空間上の一点 q と領域 P に関する最小距離 $\text{MINDIST}(q, P)$, 最大距離 $\text{MAXDIST}(q, P)$ を以下のように定義する。これらの距離尺度は, 木構造索引上の k 近傍探索で利用される距離尺度 [3], [4] と同一と考えてよい。

$$\text{MINDIST}(q, P) \equiv \min_{p \in P} \{\|\overline{qp}\|\}$$

$$\text{MAXDIST}(q, P) \equiv \max_{p \in P} \{\|\overline{qp}\|\}$$

ただし, 開線分の長さ $\|\overline{xy}\|$ は端点 x, y 間のユークリッド距離によって与えられる量である。

また, 空間オブジェクト間の遮蔽関係の判定には, 問い合わせ点 q を中心とする超球表面上への領域の射影図形を考えることが有用である。この射影図形を領域のイメージと呼ぶ。

[定義 3] (イメージ) \mathbb{R}^d 上の点 q と領域 P が与えられたとき, q を始点とし, 点 $p \in P$ を通る半直線 \overrightarrow{qp} と q を中心とする d 次元単位超球の表面との共通部分全体からなる領域を q に対する領域 P のイメージと呼び, $\pi_q P$ と表す。

点と領域に関する距離尺度およびイメージと遮蔽関係の間には, 以下に示す重要な関係が存在する (図 1)。

[補題 2] 点 q および領域 O, P に対し,

$$\text{MAXDIST}(q, O) < \text{MINDIST}(q, P) \text{ かつ}$$

$$\pi_q P \subset \pi_q O \text{ ならば } q \text{ から見て } O \text{ は } P \text{ を隠す。}$$

証明: 領域 P 中の任意の点 p について考える。定義より $\pi_q p \in \pi_q P \subset \pi_q O$ であるので点 $o \in O$ が存在し, $\pi_q o = \pi_q p$ である。すなわち, o, p は半直線 $\overrightarrow{q\pi_q p}$ 上の点である。また, $\text{MAXDIST}(q, O) < \text{MINDIST}(q, P)$

より $\|\overline{qo}\| < \|\overline{qp}\|$ である．したがって点 o は開線分 \overline{qp} 上の点である． □

4. 木構造空間索引を用いた可視探索アルゴリズム

本研究では空間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点始点可視探索を考える．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．我々はデータの追加，削除に柔軟に対応するために，空間データベースシステムで用いられている既存の空間索引手法の利用を考える．

特に本稿では，R 木 [2] またはその変種の利用を想定し，以下に示す特徴を持つ木構造索引を利用して索引付けされた空間データを対象とするアルゴリズムを考える．

(1) 木構造は非葉ノード，葉ノードから構成される．各ノードは複数のエントリから構成される．

(2) 葉ノードのエントリには 1 個の空間オブジェクト P の領域情報 $S(P)$ が格納される．従って，葉ノードのエントリには広がりを持つ領域を格納可能でなくてはならない．各空間オブジェクトは高々 1 個のエントリに格納される．

(3) 非葉ノードのエントリには 1 個の子ノード C への参照と C が対応する領域の情報 $S(C)$ が格納される．

(4) 木構造の各ノードは，そのノードのエントリが持つ領域全てを完全に包含する領域と対応する．

このような特徴を持つ木構造索引を構築可能である索引手法としては，R 木およびその変種の他に skd 木 [10] などがある．

以下に木構造空間索引上の空間走査手法を利用した可視探索アルゴリズムの概略を示す．

提案アルゴリズムは幾何問題の解決に広く利用される平面走査法の拡張である空間走査手法を利用する．通常の平面走査法と本アルゴリズムで利用する空間走査手法の違いは，平面走査法では直線により二次元空間を走査するのに対し，本アルゴリズムで利用する空間走査手法では d 次元超球の表面により d 次元空間を走査する点である．

以降では問い合わせ点 q を中心とする走査超球の表面による空間走査を考える．走査超球の半径 s はアルゴリズムの進行に伴い単調に増加する．提案アルゴリズムにおいては，探索対象オブジェクトと木構造索引のノードの双方を走査対象オブジェクトとする．提案アルゴリズムは走査対象オブジェクト集合上を走査超球面によって

走査しながら不可視オブジェクトの除去による探索空間の縮小を行う．

アルゴリズムでは平面走査法と同様の二種のデータ構造を用いる．1 つは探索空間 \mathbb{R}^d 中のイベント点と一対一に対応するイベントを保持するデータ構造 E であり，イベント計画と呼ばれる．もう 1 つはある走査超球の状態に対し，その外部に位置する不可視オブジェクトの発見に利用されるデータ構造 S であり，走査線計画と呼ばれる．

イベント計画 E に保持されるイベントは，走査対象オブジェクト P の最近点イベント e_p^r または最遠点イベント e_p^f である．アルゴリズム中では，探索対象オブジェクトに対しては最近点イベントと最遠点イベントを，木構造索引のノードに対しては最近点イベントを考える．

最近点イベント e_p^r は走査対象オブジェクト P の問い合わせ点 q への最近点と一対一に対応する．空間オブジェクト P の q への最近点とは， $s \in S(P)$ かつ $\|\overline{qs}\| = \text{MINDIST}(q, S(P))$ なる点のうちの一つである．最近点イベント e_p^r は木構造索引中での P の上位ノード N の最近点イベント e_N^r の処理に伴ってイベント計画に挿入され，走査超球の半径 s が $s = \text{MINDIST}(q, S(P))$ となったときに処理される．

探索対象オブジェクト P の最遠点とは， $s \in S(P)$ かつ $\|\overline{qs}\| = \text{MAXDIST}(q, S(P))$ なる点のうちの一つである．最遠点イベント e_p^f は探索対象オブジェクト P の最近点イベント e_p^r の処理に伴ってイベント計画に挿入され，走査超球の半径 s が $s = \text{MAXDIST}(q, S(P))$ となったときに処理される．

半径 s の走査超球に対し，

$$\mathbb{O}(s) = \{O \mid \text{MAXDIST}(q, S(O)) < s, O \in \mathbb{P}\}$$

とする． $\mathbb{O}(s)$ は走査超球内に完全に包含される探索対象オブジェクト全体からなる集合である．アルゴリズムの進行中，走査超球の半径が s の時点での $\bigcup_{O \in \mathbb{O}(s)} \pi_q S(O)$ として得られる領域を S とし，この領域を表す構造を走査線計画とする． $S(P)$ のイメージは P の最遠点イベント e_p^f の処理に伴い走査線計画 S に加えられる．

提案アルゴリズムは，他の多くの空間探索アルゴリズムと同様に多段ステップ問い合わせ処理のフィルタリングステップ [11] の実現アルゴリズムである．すなわち，提案アルゴリズムによって得られる空間オブジェクト集合は q からの可視オブジェクト集合 \mathbb{V} を完全に包含する空間オブジェクト集合 \mathbb{V}' である．提案アルゴリズムにより得られた可視オブジェクト候補集合 \mathbb{V}' から真に可視で

あるオブジェクト集合 \mathbb{V} を得るには、 \mathbb{V}' に対し既存の可視探索アルゴリズムをリファインメントステップとして適用する。 \mathbb{V}' のサイズが探索対象データ \mathbb{P} と比較して十分小さければ、可視探索全体に要する計算時間を削減可能であると考えられる。

[アルゴリズム 1](VFP フィルタリング) 入力：問い合わせ点 q , 木構造空間索引の根ノード R , 出力：可視オブジェクト候補集合 \mathbb{V}'

step 1.) 走査線計画 S , イベント計画 E を空集合とする。

step 2.) イベント計画 E に木構造索引の根ノード R の最近点イベント e_R^n を追加する。

step 3.) イベント計画 E が空でない間、以下の step 3-a, 3-b を繰り返す。

step 3-a.) イベント計画 E から問い合わせ点 q に最も近いイベント e を取り出す。

step 3-b.) イベント e の種類に応じて下記に示すイベント処理 (step n1 ~ n4 または step f1 ~ f3) を行う。

最近点イベント処理：

step n1.) 取得した最近点イベントを e_P^n とする。

step n2.) $\pi_q S(P) \subset S$ であるならば、何もせずイベント処理を終了する。

step n3.) P が探索対象オブジェクトならば最遠点イベント e_P^f をイベント計画 E に追加し、 P を可視オブジェクト候補集合 \mathbb{V}' に追加する。

step n4.) P が索引木のノードならば P の全てのエン트리 C に対して以下の step n4-a を行う。

step n4-a.) $\pi_q S(C) \not\subset S$ であるならば、 C の最近点イベント e_C^n をイベント計画 E に追加する。

最遠点イベント処理：

step f1.) 取得した最遠点イベントを e_P^f とする。

step f2.) $\pi_q S(P)$ を走査線計画 S に追加する。

アルゴリズム中において、イベント計画 E のデータ構造は優先待ち行列により自然に実現可能である。優先待ち行列内では、 q との距離が小さいイベント点に対応するイベントほど高い優先順位を持つとする。また、走査線計画 S のデータ構造としては、 $d-1$ 次元の木構造空間索引が利用可能である。イメージ間の包含判定は木構造索引を利用した範囲検索により解決される。

[定理 1] VFP フィルタリングアルゴリズムにより得られる可視オブジェクト候補集合 \mathbb{V}' は可視オブジェクト集合 \mathbb{V} を完全に包含する。

証明： 木構造索引の根ノード R の最近点イベントは明らかにイベント計画 E に追加される。

いま、最近点イベント e_P^n がイベント計画 E 中に追加されたとする。

P が木構造索引のノードである場合、 P の任意のエントリ C について考える。仮定より木構造の各ノードは、そのノードのエントリが持つ領域全てを完全に包含する領域と対応する。従って補題 1 より C の子孫中に q から可視である走査対象オブジェクトが存在するならば C および P は q から可視である。

補題 2 より P, C が q から可視ならば $\pi_q S(P) \not\subset S, \pi_q S(C) \not\subset S$ であるので、 P について step n4-a が実行され、 e_C^n がイベント計画 E に追加される。従って、 P のエントリのうち、可視である子孫を持つ全てのノードの最近点イベントは E に追加される。

以上より可視であるオブジェクトの最近点イベントは必ずイベント計画 E に追加される。

一方、 P が探索対象オブジェクトである場合、補題 2 より P が q から可視ならば $\pi_q S(P) \not\subset S$ であるので、 P は \mathbb{V}' に追加される。

従って、アルゴリズム終了の時点で $\mathbb{V} \subset \mathbb{V}'$ である。 □

5. 評価

提案手法による可視探索の性能について考察するため、上記のアルゴリズムを実装し実験の評価を行った。実験においては、二次元ユークリッド空間上に人工的に生成した凸多角形の集合に対し、提案手法および回転式平面走査による手法 [12] の二種類の手法で可視探索を行い、実行時間およびディスク入出力の回数を測定した。

実験は主記憶量 512MB の 2.4GHz Pentium4 プロセッサマシン上で行った。ディスクのページサイズは 4096 バイトとし、10 ページ相当をディスク入出力専用のバッファとして主記憶中に確保した。提案手法の実装にあたっては、Hadjieleftheriou らによる空間索引構築基盤ライブラリ [13] によって提供される R^* 木を利用した。 R^* 木生成パラメータの値は文献 [14] に従った。また、提案手法は多段問い合わせ処理のフィルタリングステップであるため、抽出された可視オブジェクト候補集合中から真に可視であるオブジェクト集合を抽出するためのリファインメントステップとして回転式平面走査による手法を利用した。

評価実験では、検索対象オブジェクト数が 10000 ~ 100000 であるような合成データ集合を合計 100 セット作成し、それらの各々について 10 個の問い合わせ点による可視探索を行った。

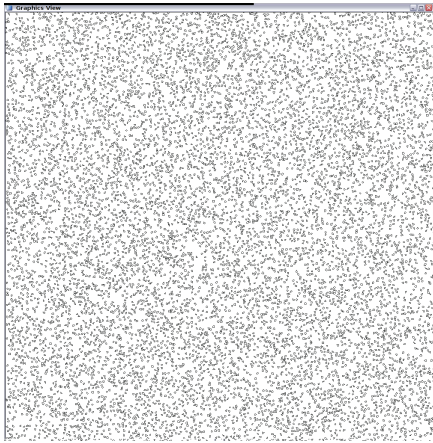


図 2 実験用合成データ例

Fig. 2 An example of synthetic data.

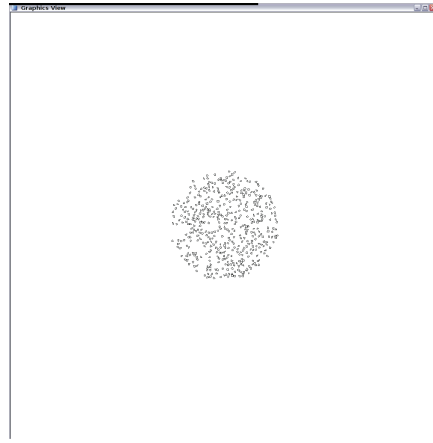


図 4 提案手法により得られた可視オブジェクト候補集合

Fig. 4 Visible candidates obtained by our method.



図 3 点 (0.5,0.5) を問い合わせ点とする可視オブジェクト集合

Fig. 3 Visible objects from (0.5, 0.5).

実験に使用した合成データの例を図 2 に示す。実験で使用した合成データは単位二次元空間 $0 \leq x \leq 1, 0 \leq y \leq 1$ 上に一様乱数に従って生成した平均頂点数約 8.9 個の互いに重なり合わない凸多角形からなる集合であり、凸多角形内部の占める面積の総和は約 0.16 であった。図 2 の例では凸多角形の数は 10000 である。

図 2 の例に対し、領域中央の点 (0.5,0.5) を問い合わせ点とする可視オブジェクト集合と提案手法で得られた可視オブジェクト候補集合を図 3 および図 4 に示す。この例では、実際の可視オブジェクトの数は 73 個であったのに対し、提案手法で得られた候補オブジェクト数は 653 個であった。

提案手法と回転式平面走査法の組み合わせによる可視探索に要する計算時間およびディスク入出力数の評価結果を図 5 および図 6 に示す。図中において、“Spherical

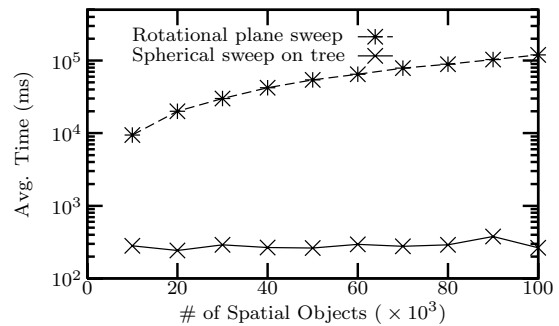


図 5 可視探索実行時間の比較

Fig. 5 Time spent in visibility search algorithms.

sweep on tree” は、提案手法によるフィルタリングと回転式平面走査によるリファインメントの組み合わせによる可視探索を、“Rotational plane sweep” は回転式平面走査法単独での可視探索を示す。

図より、提案手法を利用した可視探索では、既存手法を単独で用いた可視探索と比較して 10 分の 1 未満の計算時間とディスク入出力数で可視探索が可能となることが確認される。また、既存手法のみを単独で用いた探索では、探索対象オブジェクト数の増大に従って計算時間およびディスク入出力数も増大しているのに対し、提案手法を利用した探索では実行時間、ディスク入出力数の変化は小さいことがわかる。

提案手法の出力感応性の評価結果を図 7 に示す。図より提案手法のディスク入出力数は可視オブジェクト候補の数の依存することが推測される。これら二つの変量の間の相関係数は約 0.871 であった。この結果は、本手法を利用した場合、同一の探索対象オブジェクト集合に対する同一の問い合わせ点からの可視探索であっても、木

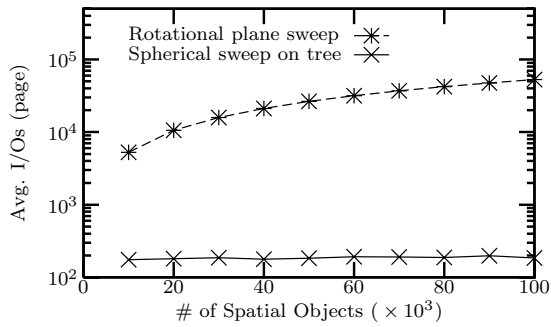


図 6 ディスク入出力数の比較

Fig. 6 I/O in visibility search algorithms.

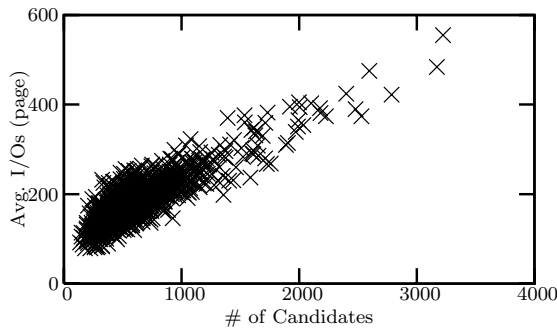


図 7 ディスク入出力数の可視オブジェクト候補数への感応性

Fig. 7 I/O Sensitivity to number of candidates.

構造空間索引の構成法により可視探索性能が大きく影響されることを示唆している。

6. ま と め

本稿では、木構造索引を利用して空間データベースに格納された空間オブジェクト集合を対象とする可視探索手法を提案し、検索性能の実験的評価を行った。

提案手法では既存の空間データベースシステムで用いられている R 木などの木構造索引をそのまま可視探索に利用可能であり、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、二次元ユークリッド空間上に生成した凸多角形のデータ集合に対し、提案手法を用いた可視探索を行った際の計算時間とディスク入出力数を計測した。実験結果から、提案手法は既存手法の一つである回転式平面走査法と比較し優位であることが確認された。

今回の実装評価においては、応用分野の特性を利用しない古典的可視探索手法の一つである回転式平面走査法を比較対象とした。しかし実用の観点からは各種の応用上の特性を利用するヒューリスティクスに基づく手法との性能比較が必要であると考えられる。同時に、実際の

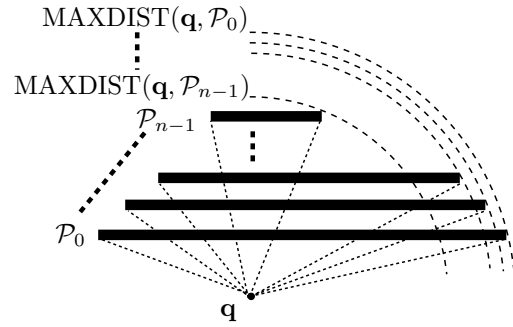


図 8 本手法で探索空間縮小が行われない例

Fig. 8 In this case, our method accesses all objects.

応用分野で用いられるデータセットを用いた、提案手法のより詳細な評価も今後の課題である。

また、提案手法においては空間オブジェクトの単位球面への射影に伴う計算量の増大と数値計算誤差の発生がある。この問題は、単位球面への射影を有理数演算内で処理可能である適切な射影演算で代替することにより解決可能であると考えられる。

さらに、特定の性質を持った探索対象オブジェクト集合と問い合わせ点の組み合わせに対しては、提案手法によって十分に小さな可視オブジェクト候補集合を得ることができないことがわかっている。図 8 はそのような探索対象オブジェクト集合の例である。この図のように $\text{MAXDIST}(q, \mathcal{O}) < \text{MINDIST}(q, \mathcal{P})$ であるような空間オブジェクトの \mathcal{O}, \mathcal{P} の組が探索対象オブジェクト集合中に存在しない場合、提案手法では探索範囲の限定は行われない。実用上はこのような探索対象オブジェクト集合が与えられる可能性は小さいと考えられるが、空間データベースシステムで汎用の用途に供すべき検索手法にとって、入力によって極度に性能の劣化が発生するという性質は望ましいものではない。どのような入力に対しても適切な性能を期待できるアルゴリズムの構築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] 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).
- [3] N. Roussopoulos, S. Kelley and F. Vincent: "Nearest neighbor queries", Proc. of ACM SIGMOD Intl. Conf. on Management of Data, pp. 71–79 (1995).
- [4] N. Katayama and S. Satoh: "The SR-tree: an index structure for high-dimensional nearest neighbor queries", Proc. of ACM SIGMOD Intl. Conf. on Man-

- agement of Data, pp. 369–380 (1997).
- [5] N. Greene, M. Kass and G. Miller: “Hierarchical Z-buffer visibility”, SIGGRAPH ’93: Proc. of the 20th annual Conf. on Computer graphics and interactive techniques, pp. 231–238 (1993).
 - [6] H. Fuchs, Z. M. Kedem and B. F. Naylor: “On visible surface generation by a priori tree structures”, SIGGRAPH ’80: Proc. of the 7th annual Conf. on Computer graphics and interactive techniques, pp. 124–133 (1980).
 - [7] D. Gordon and S. Chen: “Front-to-back display of BSP trees”, IEEE Computer Graphics and Applications, **11**, 5, pp. 79–85 (1991).
 - [8] S. Coorg and S. Teller: “Real-time occlusion culling for models with large occluders”, SI3D ’97: Proc. of Symp. on Interactive 3D graphics, New York, NY, USA, ACM Press, pp. 83–ff. (1997).
 - [9] T. Hudson, D. Manocha, J. Cohen, M. Lin, K. Hoff and H. Zhang: “Accelerated occlusion culling using shadow frusta”, SCG ’97: Proc. of the 13th annual Symp. on Computational geometry, New York, NY, USA, ACM Press, pp. 1–10 (1997).
 - [10] B. C. Ooi: “Efficient Query Processing in Geographic Information Systems”, Vol. 471 of Lecture Notes in Computer Science, Springer-Verlag (1990).
 - [11] T. Brinkhoff, H.-P. Kriegel, R. Schneider and B. Seeger: “Multi-step processing of spatial joins”, Proc. of ACM SIGMOD Intl. Conf. on Management of Data, pp. 197–208 (1994).
 - [12] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf: “Computational Geometry Algorithms and Applications”, Springer-Verlag (1997).
 - [13] M. Hadjieleftheriou, E. Hoel and V. Tsotras: “SaLL: 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).
 - [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).