

空間データベースシステムのための 可視探索手法 Visibility Search for Spatial Database Systems

能登谷 淳一* 杉本 雄太†
草苅 良至‡ 笠井 雅夫§

Junichi NOTOYA Yuta SUGIMOTO
Yoshiyuki KUSAKARI Masao KASAI

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 CG 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見出す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。

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. 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.

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り、広い応用範囲を持つ幾何探索問題の一つである。本研究では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。

本研究で提案するフィルタリング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。

2. 関連研究

可視探索問題は、従来コンピュータグラフィックス (CG) の分野で着目されてきた問題である。CG の分野では可視探索問題の三次元レイトレーシングへの応用に関する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値の計算である。そのため、従来の研究では、各画素の中心を通る半直線の可視探索が主要な研究テーマであった。この問題に対しては、z バッファ法やその発展アルゴリズム [2], BSP 木

を利用した可視探索の高速化 [3] などの研究が行われている。

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、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 [4], [5] が提案されている。

これらの CG 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
- 主記憶に格納可能な量のデータを対象とする
- 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
- 二次記憶上の空間オブジェクトデータは可視探索以外の用途にも利用される
- データの更新が頻繁に発生する
- 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、遮蔽関係に基づく探索空間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

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。

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 $S(P) \subset \mathbb{R}^d$ と対応する。 $S(P)$ を空間オブジェクト P の領域と呼ぶ。

本研究で着目する可視探索問題は、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] 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、以下の関係が存在する。

[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。□

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。

* 正会員 秋田県立大学システム科学技術学部 notoya@akita-pu.ac.jp

† 秋田県立大学システム科学技術研究所 m07b007@akita-pu.ac.jp

‡ 秋田県立大学システム科学技術学部 kusakari@akita-pu.ac.jp

§ 秋田県立大学システム科学技術学部 kasai@akita-pu.ac.jp

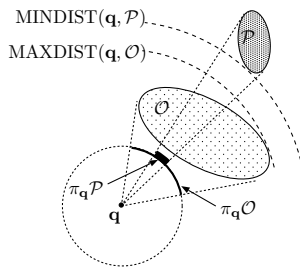


図1 領域の距離とイメージを利用した遮蔽関係の判定
Fig.1 Occluding determination by distance metrics and projected images of regions.

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

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

ただし、開線分の長さ $\|xy\|$ 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] \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 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、以下の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、 $\text{MAXDIST}(q, O) < \text{MINDIST}(q, P)$ かつ $\pi_q P \subset \pi_q O$ ならば q から見て O は P 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 $q\pi_q p$ 上の点である。また、 $\text{MAXDIST}(q, O) < \text{MINDIST}(q, P)$ より $\|qo\| < \|qp\|$ である。したがって点 o は開線分 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、データの更新が頻繁に発生する。我々はデータの更新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木およびその変種[1]の利用を想定し、以下に示す特徴を持つ木構造索引に格納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。

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。

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イベント計画 E であり、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走査線計画 S である。

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^n 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^n 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 $\|qs\| = \text{MINDIST}(q, S(P))$ なる点のうちの一つである。最近点イベント e_P^n は木構造索引中の P の上位ノード N の最近点イベント e_N^n 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 $\|qs\| = \text{MAXDIST}(q, S(P))$ 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^n 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、

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

とする。 $O(s)$ 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 $\bigcup_{O \in O(s)} \pi_q S(O)$ 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[6]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 \mathbb{V}' 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。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.) E に R の最近点イベント e_R^n を追加する。
- step 3.) E が空でない間、以下の step 3-a, 3-b を繰り返す。
 - step 3-a.) E から q に最も近いイベント e を取り出す。
 - step 3-b.) 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.) $\pi_q S(P) \subset S$ 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 E に追加し、 P を \mathbb{V}' に追加する。
- step n4.) P が索引木のノードならば P の全てのエントリ C に対して以下の step n4-a を行う。

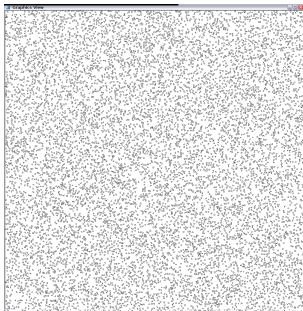


図2 実験用合成データ例
Fig.2 An example of synthetic data.

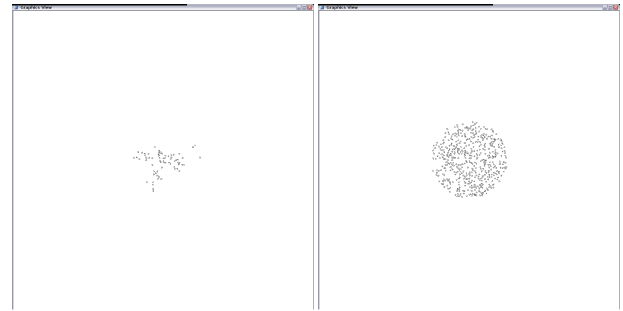


図3 可視オブジェクト集合と可視オブジェクト候補
Fig.3 Visible objects and visible candidates.

step n4-a.) $\pi_q S(C) \not\subset S$ ならば, C の最近点イベント e_C^n を 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 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 \mathcal{V}' は可視オブジェクト集合 \mathcal{V} を完全に包含する.

証明: R の最近点イベントは明らかに E に追加される.

いま, 最近点イベント e_P^n が 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 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 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, $\pi_q S(P) \not\subset S$ より P は \mathcal{V}' に追加される. 従って, アルゴリズム終了の時点で $\mathcal{V} \subset \mathcal{V}'$ である. \square

5. 評価

提案手法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的評価を行った. 実験では, 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 [7] 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 [8] 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 [9] に従った. また, リファインメントステップとしては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.

実験に使用した合成データの例を図 2 に示す. 合成データは

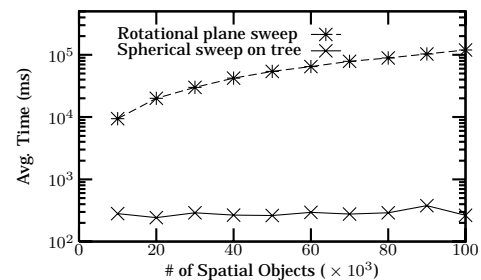


図4 可視探索実行時間の比較
Fig.4 Time spent in visibility search algorithms.

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 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 4 および図 5 に示す. 図中において, “Spherical 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ってい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 6 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に依存することが推測される. これら二つの変数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

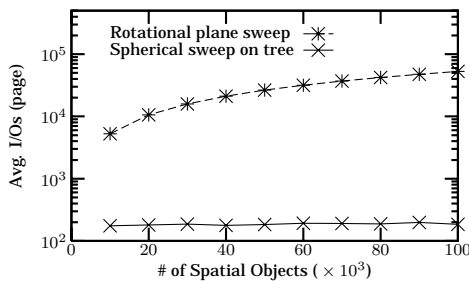


図5 ディスク入出力数の比較
Fig. 5 I/O in visibility search algorithms.

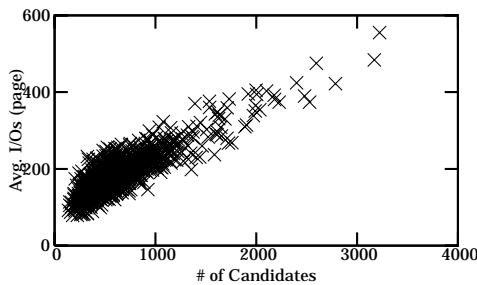


図6 ディスク入出力数の可視オブジェクト候補数への感応性
Fig. 6 I/O Sensitivity to number of candidates.

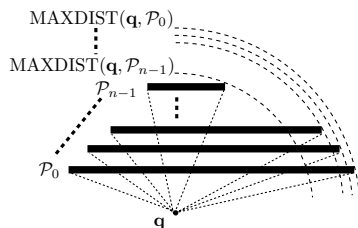


図7 本手法で探索空間縮小が行われない例
Fig. 7 In this case, our method accesses all objects.

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7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 $\text{MAXDIST}(q, O) < \text{MINDIST}(q, P)$ であるような空間オブジェクト O, 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ぬ。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] 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).
- [3] D. Gordon and S. Chen: "Front-to-back display of BSP trees", IEEE Computer Graphics and Applications, **11**, 5, pp. 79-85 (1991).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf: "Computational Geometry Algorithms and Applications", Springer-Verlag (1997).
- [8] 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).
- [9] 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).

能登谷 淳一 Junichi NOTOYA

秋田県立大学システム科学技術学部電子情報システム学科助手。1999 筑波大学大学院博士課程工学研究科修了。博士(工学)。データベースシステムの研究・開発に従事。情報処理学会会員。電子情報通信学会会員。日本データベース学会会員。

杉本 雄太 Yuta SUGIMOTO

秋田県立大学大学院システム科学技術研究科博士前期課程在学中。2005 秋田県立大学システム科学技術学部電子情報システム学科卒業。計算幾何学、アルゴリズムの研究に従事。

草薙 良至 Yoshiyuki KUSAKARI

秋田県立大学システム科学技術学部電子情報システム学科助教授。1998 東北大学大学院情報科学研究科博士課程修了。博士(情報科学)。計算幾何学、アルゴリズムの研究に従事。

笠井 雅夫 Masao KASAI

秋田県立大学システム科学技術学部電子情報システム学科教授。1975 東京大学大学院工学系研究科博士課程修了。工学博士。保全合理化・最適化問題等の研究に従事。