

3次元TINデータ上での空間的スカイライン問合せ Spatial Skyline Queries on 3D TIN Data

笠井 雄太
Yuta Kasai

杉浦 健人
Kento Sugiura

石川 佳治
Yoshiharu Ishikawa

1 はじめに

意思決定の支援にはスカイライン問合せが有用である。スカイライン問合せとは、データの集合が与えられた際に他のデータに支配されない集合を検索する問合せである [1]。例えば海の近くにある安いホテルや現在地から近い評価の高い食事店など、対象データが複数の評価指標を持つ際に有効であり、スカイライン問合せを解くことで意思決定における適切な候補を列挙できる [2]。

スカイライン問合せを地理空間上に拡張した研究として、Sharifzadeh らは空間的スカイライン問合せ (*spatial skyline query*) を提案した [3]。空間的スカイライン問合せでは、空間上に存在する複数人からの距離を評価指標としてスカイライン問合せを定義しており、例えば集合場所の決定などに応用できる。しかし、この研究では空間上の移動距離をユークリッド距離で計算しており、三次元空間における高低差の激しい土地や山を跨ぐような広範囲な地域への問合せにおいて計算上で使用する距離と実際の移動距離との差が大きいという問題がある。

一方、複雑な三次元オブジェクトを表現するためのデータモデルとして、3D TIN (*triangulated irregular network*) が注目を集め始めている。3D TIN データは三次元の頂点と辺の集合からなるグラフデータの一種であり、三角形のみのネットワークとして構成される。三次元オブジェクトを表現するデータとしては他に点群データなどがあるが、3D TIN は辺の情報を用いて物体の面を柔軟かつ効率的に表現する。つまり、3D TIN を用いることで、三次元オブジェクト上の移動についてより正確な距離を計算できる。

そこで、本稿では3D TIN 空間上における空間的スカイライン問合せを提案する。距離の計算にユークリッド距離ではなく3D TIN 上の表面距離を用いることで、より正確な移動距離に基づいた三次元空間上でのスカイライン問合せを定義する。更に、3D TIN 上での移動距離に関する索引 [4] を利用し、この問題を解く方法を提案する。

2 問題定義

$P = \{p_1, \dots, p_m\}$ を3D TIN 上の点の集合、 $Q = \{q_1, \dots, q_n\}$ を3D TIN 上の問合せ点の集合、 $D_s: \mathbf{R}^3 \times \mathbf{R}^3 \rightarrow \mathbf{R}$ を3D TIN 表面上の距離関数と定義する。 Q に関して $p_a \in P$ が $p_b \in P$ を3D TIN 上で空間的に支配する (*spatially dominate on 3D TIN*) ことを $p_a <_Q p_b$ と表記し、次の式 (1) で定める。

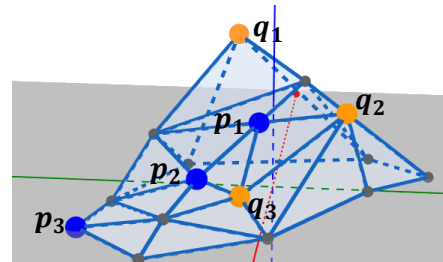


図1 3D TIN 空間上の空間的スカイライン問合せの例

$$p_a <_Q p_b \equiv \forall q \in Q, D_s(p_a, q) < D_s(p_b, q) \quad (1)$$

p が P 中のどの点にも支配されない場合、 p をスカイライン点と呼ぶ。 Q に対するスカイライン点の集合が3D TIN 空間上の空間的スカイライン問合せの解である。例えば、図1の3D TIN における空間的スカイライン問合せを考える。各問合せ点 $q \in \{q_1, q_2, q_3\}$ からの距離を計算すると全ての問合せ点において $D_s(p_2, q) < D_s(p_3, q)$ であり、 p_3 は p_2 に空間的に支配される ($p_2 <_Q p_3$)。つまり、 p_3 はスカイライン点ではない。一方、 p_1 と p_2 は問合せ点 q_2 と q_3 について互いに優位性があり ($D_s(p_1, q_2) < D_s(p_2, q_2)$ かつ $D_s(p_2, q_3) < D_s(p_1, q_3)$)、それぞれ空間的に支配されていない。したがって、スカイラインは $S(Q) = \{p_1, p_2\}$ となる。

空間的スカイライン問合せと比較して3D TIN を用いた空間的スカイライン問合せではより大きな計算時間が必要である。3D TIN を構成する頂点数を N として、二点間の距離 $D_s(\cdot, \cdot)$ を求めるための時間計算量は $\Theta(N^2)$ である [5]。これを元に式 (1) を用いて愚直な手法でスカイライン点を判定する時間計算量は $\Theta(N^2|P|^2|Q|)$ となる。

3 空間的スカイライン問合せとの比較

Sharifzadeh らは空間的スカイライン問合せを定義するとともにその効率的な解法も提案しており、解の集合を $S(Q)$ 、問合せ点集合 Q の凸包を $CH_r(Q)$ と定義したときの時間計算量を $\Theta(|S(Q)|^2|CH_r(Q)| + \log|P|)$ に抑えた [3]。以下では、時間計算量の削減における主要なアイデアを簡潔に説明する。

候補点集合 P から一部の解 $S(Q)$ が見つけれられた後、新たな候補点 p_{new} は既に判明している解に含まれる点 $p_j \in S(Q)$ に対して次の三つの状態に分類される。

1. p_j に支配される ($\exists p_j \in S(Q), p_j <_Q p_{new}$)。
2. p_j を支配する ($\exists p_j \in S(Q), p_{new} <_Q p_j$)。
3. それ以外 ($\forall p_j \in S(Q), p_{new} \not<_Q p_j \wedge p_j \not<_Q p_{new}$)。

このうち2.ないし3.となった時点で p_{new} は解 $S(Q)$ に含まれるため、1.の状態になるかならないかを効率的に判定できればよい。これに対し Sharifzadeh らは、各問合せ点 $q \in Q$ を中心とした半径 $D(q, p_j)$ の円の領域内に p_{new} が含まれないとき p_{new} が p_j によって支配される (1.の状態になる) ことを導いた。つまり、スカイライン問合せにおける支配関係の判定を、ある領域に対する空間的な交差関係の判定へと帰着させた。 p_{new} が支配されるかどうかの判定には通常 $\Theta(|P||Q|)$ の時間計算量が必要であるが、円の和集合に対する最小包含長方形に点 p_{new} が含まれるかの判定は $\Theta(1)$ で実行できるため、空間的スカイライン問合せの解の効率的な検出を実現している。

しかし、この計算量の削減はユークリッド空間における距離の使用を前提としており、3D TIN 上の表面距離では成立しない。具体的には、空間的な交差判定に点集合 P から生成したボロノイ図及び問合せ点集合 Q の凸包を利用しているが、3D TIN 空間上におけるボロノイ図や凸包の定義は明らかではない。つまり、既存手法を単純に3D TIN へと拡張することは難しい。更に、3D TIN 上の表面距離 D_s の計算には $\Theta(N^2)$ の時間計算量が必要であり、空間的な交差判定を用いない方法に必要な計算量はユークリッド空間における空間的スカイライン問合せよりも大きいという問題もある。

4 計算時間を減らすための手法

3D TIN 上における空間的スカイライン問合せの問題の一つは、表面距離の計算に必要な時間計算量の大きさである。そこで、より高速に計算できるユークリッド・ネットワーク距離の利用を考える。なお、本稿におけるネットワーク距離とは3D TIN 上の辺のみを経路として使用する場合の距離を示す。各問合せ点 $q \in Q$ から3D TIN 上の任意の頂点へのネットワーク距離は時間計算量 $\Theta(|Q|N \log N)$ で事前計算できるため、処理時には $\Theta(1)$ で使用できる。

ユークリッド距離を D_e 、表面距離を D_s 、ネットワーク距離を D_n で表すとき、以下の関係が成り立つ [4]。

$$D_e(s, t) \leq D_s(s, t) \leq D_n(s, t) \quad (2)$$

この式から、点 $p, p' \in P$ と問合せ点 $q \in Q$ について下記の関係が成り立つ。

$$D_n(q, p) < D_e(q, p') \Rightarrow D_s(q, p) < D_s(q, p') \quad (3)$$

つまり、式 (1) の空間的な支配関係の一部はユークリッド距離とネットワーク距離から求められる。

$$\forall q \in Q, D_n(q, p) < D_e(q, p') \Rightarrow p <_Q p' \quad (4)$$

4.1 最小包含長方形の変形による判定の簡略化

3章で述べたとおり、三次元空間では半径 $D_e(q, p_j)$ の円の領域内に p_{new} が含まれない時、 p_{new} が p_j によって支配されるため p_{new} はスカイライン点ではないのであった。3D TIN 上で同じ手法を採る場合、半径 $D_s(q, p_j)$ の円を用いる必要があるが、式 (4) を用いることで D_s を用いずに p_{new} がスカイライン点ではないことを Sharifzadeh らの方法と同様に判定できる。より具体的には、各問合せ点 $q \in Q$ を中心とした半径

$D_n(q_i, p_j)$ の円を作成しておき、円の和集合に対する最小包含長方形の内に点 p_{new} が含まれるかを判定すれば空間的スカイライン問合せと同様の方法で $\Theta(1)$ の判定が可能である。

4.2 擬似的なボロノイ図の作成

既存手法で用いられたボロノイ図の代替として、3D TIN 上における索引を用いる手法について述べる。候補点の集合 P に対する3D TIN 上のボロノイ図を構築するための時間計算量は $\Theta(|P|N^3)$ [5] であり、単純に3D TIN 上の空間的スカイライン問合せを解くよりも計算量が大きい。そこで、Shahabi ら [4] によって提案された TSI (tight surface index) 及び LSI (loose surface index) の利用を考える。

TSI は3D TIN のモデル T 上の各点 $p \in P$ に対して以下の式で定義され、候補点がスカイライン点であることの判定に利用できる。

$$TSI(p) \equiv \{r \mid r \in T \wedge \forall p' \in P \setminus \{p\}, D_n(p, r) < D_e(p', r)\} \quad (5)$$

中心が点 $p, p' \in P$ であるボロノイセルの境界線をなす点の集合は $D_s(p, m) = D_s(p', m)$ をみたす点 m の集合であるため、TSI はボロノイセルよりも狭い範囲を表現している。TSI を用いることである問合せ点 $q_i \in Q$ があるボロノイセル $VC(p_j)$ に含まれるかを判定し、式 (1) より即座に候補点 p' がスカイライン点とできる。

一方、LSI は各点 $p \in P$ に対して以下の式で定義され、候補点がスカイライン点でないことの判定に利用できる。

$$LSI(p) \equiv \{r \mid r \in T \wedge \forall p' \in P \setminus \{p\}, D_e(p, r) < D_n(p', r)\} \quad (6)$$

LSI はボロノイセルよりも広い範囲を表現しており、ボロノイセルに確実に含まれない点を判定できる。つまり、LSI に含まれないとき、その点はスカイライン点ではない。

したがって、スカイライン点であるかの厳密な確認は、LSI に含まれかつ TSI に含まれない場合にのみ限定できる。

5 まとめと今後の展望

本稿では3D TIN を用いた空間的スカイライン問合せを実行するための方法及び改善点を提案した。今後は計算量の厳密な評価及び解を得るための実装と実行速度の測定を行う。

謝辞

本研究は JSPS 科研費 (JP16H01722, JP19K21530, JP20K19804) の助成及び国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務による。

参考文献

- [1] S. Borzsony, D. Kossmann, and K. Stocker, "The skyline operator," in *Proc. ICDE*, pp. 421–430, 2001.
- [2] C. Kalyvas and T. Tzouramanis, "A survey of skyline query processing," *CoRR*, vol. abs/1704.01788, 2017.
- [3] M. Sharifzadeh and C. Shahabi, "The spatial skyline queries," in *Proc. VLDB*, pp. 751–762, 2006.
- [4] C. Shahabi, L. Tang, and S. Xing, "Indexing land surface for efficient kNN query," *PVLDB*, vol. 1, no. 1, pp. 1020–1031, 2008.
- [5] J. Chen and Y. Han, "Shortest paths on a polyhedron," in *Proc. Annual Symposium on Computational Geometry (SoCG)*, pp. 360–369, 1990.