

空間ネットワーク環境における問合せ処理法

代表研究者 後藤 佑介 岡山大学 大学院自然科学研究科 准教授
共同研究者 江原 康生 京都大学 学術情報メディアセンター 特定准教授

1 はじめに

近年、Global Positioning System (GPS) を用いて目的地検索やカーナビゲーションを行う位置情報サービスが注目されている。位置情報サービスでは、空間ネットワーク上に存在するすべてのオブジェクトについて、クエリオブジェクト(以下、クエリ)が自身からもっとも近いオブジェクトを探索する最近傍探索を行う。既存の最近傍探索手法は、クエリからすべてのオブジェクトまでの距離をそれぞれ算出する必要があり、計算量は膨大になる。この計算量を削減するため、空間ネットワーク上の領域をオブジェクトごとに分割する手法(以下、領域分割手法)が提案されている。既存の領域分割手法では、各オブジェクトがクエリをもっとも近いオブジェクトと判断する領域をすべてのオブジェクトに対して、事前に作成する。クエリは、どのオブジェクトの領域内にあるかを判別するだけで、自身の最近傍となるオブジェクトを探索でき、計算量を削減できる。しかし、既存の領域分割手法は、空間ネットワーク上でオブジェクトの位置情報が変化する場合を考慮していない。このため、空間ネットワーク上でオブジェクトが追加もしくは削除されると、オブジェクト間の位置関係が変化して、すべてのオブジェクトに対して領域の再計算が必要になる。本研究では、空間ネットワーク上における最近傍探索において、オブジェクトの追加や削除によるオブジェクト間の動的な位置関係を考慮した領域分割手法を提案し、既存手法との比較評価を行った。提案手法では、オブジェクトの追加もしくは削除で領域の作成に影響を与えるオブジェクトの領域のみを更新する。評価の結果、提案手法は、領域の更新で発生する計算量を既存手法と比較して削減できることを確認した。

2 探索手法

2-1 最近傍探索

最近傍探索では、空間ネットワーク上に複数のオブジェクトが存在する場合、クエリからの距離がもっとも近いオブジェクトを探索する。単純な最近傍探索手法では、クエリと各オブジェクト間の距離をすべて計算して比較する。この場合、空間ネットワーク上に存在するオブジェクトの数が多くなると、クエリとオブジェクト間の距離を計算する回数が多くなり、計算量は増加する。

空間ネットワークにおけるオブジェクト間の測定距離は、二つのオブジェクト間の直線距離であるユークリッド距離と、複数のオブジェクトで構成されるネットワーク経路に沿って測定するネットワーク距離の二種類に分けられる。本研究では、ユークリッド距離を用いた最近傍探索を考える。

最近傍探索の例を図1に示す。オブジェクト p_0, \dots, p_7 が空間ネットワーク上に存在し、 p_0 をクエリとして、最近傍となるオブジェクトを探索する。この場合、 p_0 からもっとも近い位置に存在するオブジェクトは p_1 であるため、 p_0 の最近傍となるオブジェクトは p_1 となる。

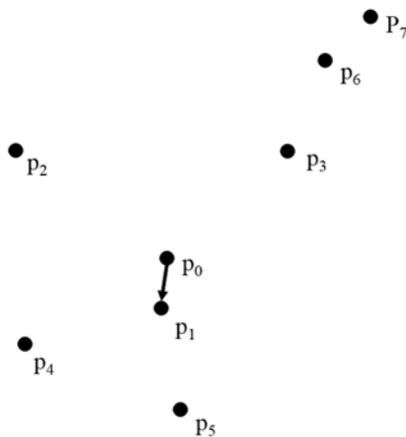


図1：最近傍探索

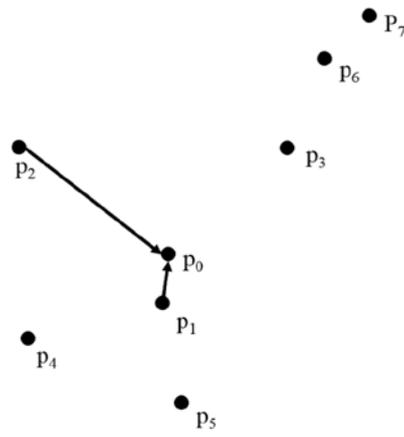


図2：逆最近傍探索

2-2 逆最近傍探索

逆最近傍探索 (RNN) [1]では、空間ネットワーク上に複数のオブジェクトが存在する場合、クエリまでの距離がもっとも近いオブジェクトを探索する。逆最近傍探索の例を図2に示す。オブジェクト p_0, \dots, p_7 が空間ネットワーク上に存在し、 p_0 をクエリとして、逆最近傍探索を行う。オブジェクト p_0, \dots, p_7 のうち、 p_0 までの距離が他のオブジェクトまでの距離より短いオブジェクトは p_1, p_2 であるため、 p_0 の逆最近傍となるオブジェクトは p_1, p_2 となる。

最近傍探索では、探索するオブジェクトが一つの場合、一つのオブジェクトが求まる。一方、逆最近傍探索では、結果は0個以上のオブジェクトとなる。例えば、図2において、 p_2 をクエリとして逆最近傍探索を行う場合、 p_2 の逆最近傍となるオブジェクトは存在しない。

3 RNN 領域の分割手法

空間ネットワーク上に存在する k ($k \geq 1$) 個のオブジェクトに対して逆最近傍探索を行うため、R木を用いた探索手法[2]、およびRkNN領域の作成手法[3]について、これまでに多くの研究が行われている。本章では、 $k=1$ の場合における効率的なRNN領域の作成手法として、2*farthest vertex法、およびContact Zoneを用いた手法の二つを順番に説明する。

3-1 2*farthest vertex 法

2*farthest vertex法[4][5]では、単純手法と同様に、空間ネットワーク上に存在するすべてのオブジェクトに対して、ターゲットから垂直二等分線を引く。複数の垂直二等分線によって多角形の領域(以下、初期領域)を作成できる場合、初期領域を構成する頂点のうちターゲットからもっとも遠い頂点の一つを選択する。次に、ターゲットと選択した頂点間の距離の2倍を半径とし、ターゲットが中心となる円を作成する。クエリは、作成した円の内側に存在するオブジェクトに対して垂直二等分線を引く。一方で、初期領域が作成されない場合、単純手法と同様に、空間ネットワーク上に存在するすべてのオブジェクトとターゲットとの間で垂直二等分線を引き、RNN領域を作成する。

3-2 Contact Zone (CZ) を用いた RNN 領域の作成手法

初めに、従来手法と同様に初期領域を作成する。次に、中心が初期領域を構成する多角形の各頂点、半径が頂点からターゲットまでの距離となるCZの円を作成し、CZの内側に存在するオブジェクトに対して、ターゲットから垂直二等分線を引く。初期領域が作成されない場合、単純手法と同様に、空間ネットワーク内に存在するすべてのオブジェクトとターゲットとの間で垂直二等分線をそれぞれ引き、RNN領域を作成する。

4 提案手法

4-1 概要

既存手法では、空間ネットワーク上でオブジェクトが追加もしくは削除されることで、オブジェクト間の位置関係が動的に変化する場合を考慮していない。この場合、すべてのオブジェクトに対してターゲットとの位置関係を再計算する必要があるため、RNN領域の更新にかかる時間は長大化する。そこで、最近傍探索において、オブジェクトの追加および削除に対応した領域分割手法を提案する。提案手法では、追加もしくは削除されたオブジェクトによってRNN領域の更新に影響するオブジェクトを判別し、RNN領域を更新することで、既存手法に比べて計算量を削減する。

4-2 想定環境

- ・ 空間ネットワークは二次元で表す。
- ・ オブジェクト間の距離はユークリッド距離で表す。
- ・ 空間ネットワーク上に存在する複数のオブジェクトからランダムに一つ選択したオブジェクトに対して、RNN領域を生成する。次に、オブジェクトの追加および削除を行い、RNN領域を更新する。

4-3 オブジェクトを追加する場合の手順

1. 空間ネットワーク上でランダムに一つ選択したオブジェクトをもとに作成したRNN領域に対して、CZとPRを作成する。

2. 追加したオブジェクトの位置に応じて、(a), (b) のうちどちらかを実行する。
 - (a) 追加したオブジェクトが「CZ の内側かつ PR の内側」、もしくは PR の外側に存在する場合、追加したオブジェクトに対してターゲットから垂直二等分線を引き、RNN 領域を更新する。
 - (b) 追加したオブジェクトが(a) 以外の領域に存在する場合、追加したオブジェクトは RNN 領域の更新に影響しないため、垂直二等分線を引かない。

オブジェクトを追加して RNN 領域を更新する前の空間ネットワークでは、ターゲット t とオブジェクト $p1, \dots, p7$ が空間ネットワーク上に存在している状態で、オブジェクト $p8$ を追加する。オブジェクト $p8$ は「Contact Zone (CZ) の内側かつ Pruning Region (PR) の内側」に存在するため、 t と $p8$ との間に垂直二等分線を引き、図 5 に、垂直二等分線を書いて RNN 領域を更新した後の例を示す。RNN 領域は、 t と $p8$ 間の垂直二等分線で更新される。

4-4 オブジェクトを削除する場合の手順

1. 空間ネットワーク上でランダムに一つ選択したオブジェクトをもとに作成した RNN 領域に対して、CZ と PR を作成する。
2. 削除したオブジェクトの位置に応じて、(a), (b) のどちらかを実行する。
 - (a) 削除したオブジェクトが PR を構成する直線上に存在する場合、 t と削除したオブジェクトとの垂直二等分線の一部となる RNN 領域の一边を削除した上で、RNN 領域で削除した辺と交差する辺を延長して、RNN 領域を更新する。次に、更新した RNN 領域に対して CZ と PR を作成し、「PR の内側かつ CZ の外側」、もしくは PR の外側にオブジェクトが存在する場合、垂直二等分線を引き。
 - (b) 削除したオブジェクトが(a) 以外の領域に存在する場合、削除したオブジェクトは RNN 領域の更新に影響しないため、垂直二等分線を引かない。

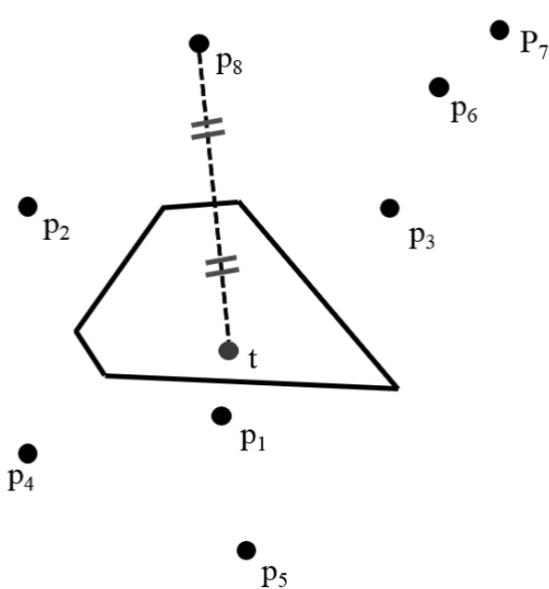


図 5 : RNN 領域例(オブジェクト追加)

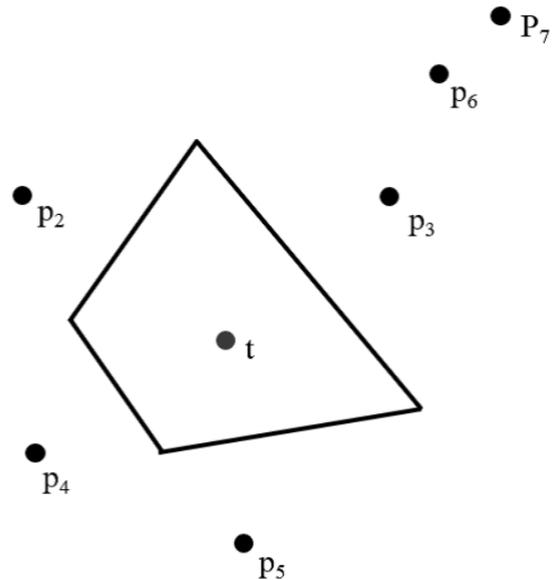


図 6 : RNN 領域例(オブジェクト削除)

空間ネットワーク上でオブジェクトを削除して RNN 領域を更新する場合、ターゲット t とオブジェクト $p1, \dots, p7$ が空間ネットワーク上に存在している状態で、オブジェクト $p1$ を削除する。オブジェクト $p1$ は PR を構成する直線上に存在するため、 t とオブジェクト $p1$ との垂直二等分線の一部となる破線で示された RNN 領域の辺を削除して、RNN 領域を更新する。次に、CZ と PR を作成した空間ネットワークについて、オブジェクト $p5$ は、色付けされた領域である PR の外側に存在するため、ターゲットとオブジェクト $p5$ との間に垂直二等分線を引き、RNN 領域を更新する。図 6 に、 t と $p5$ との間に垂直二等分線を引き、RNN 領域を更新した後の例を示す。

5 評価

5-1 概要

提案手法の有用性を検証するため、計算機上でシミュレーション評価を行った。提案手法および既存手法のプログラムは、C++で実装した。

5-2 評価環境

計算機シミュレーションによる評価環境について、以下に示す。

- ・ オブジェクト間の距離はユークリッド距離で表す。
- ・ 空間ネットワーク上のオブジェクトの数は、10, 50, 100, 500, 1000, 5000, 10000 の7種類とする。
- ・ 空間ネットワークの大きさは、縦1000, 横1000 の二次元領域とする。
- ・ 評価で用いる既存手法は、初期領域を用いずにRNN領域を再計算する単純手法を用いる。

5-3 オブジェクトを追加する場合

オブジェクトを追加する場合におけるRNN領域の更新について、提案手法および単純手法を用いて比較評価を行った。評価では、オブジェクトの追加によってRNN領域の更新にかかる処理時間を100回計測し、平均値を算出した。評価結果を図7に示す。横軸は空間ネットワーク上に存在するオブジェクトの数、縦軸はRNN領域の更新終了までの平均処理時間である。

図7より、オブジェクトの数に関係なく、提案手法の処理時間は単純手法よりも短い。提案手法では、RNN領域の更新に影響しないオブジェクトが追加された場合、垂直二等分線を引かずに処理を終了する。このため、空間ネットワーク上でランダムにオブジェクトを追加する場合、垂直二等分線を引く必要がないオブジェクトが追加される確率は大きくなり、処理時間は0に近づく。

次に、オブジェクトの追加において、RNN領域の更新に影響する領域にオブジェクトを追加した場合の評価結果では、RNN領域の更新に影響する領域に対してオブジェクトを追加した場合においても、提案手法の処理時間は単純手法より短くなる。また、提案手法では、追加されたオブジェクトに対してのみ垂直二等分線を引くため、空間ネットワーク上に存在するオブジェクトの数に関係なく、処理時間はほぼ一定となる。

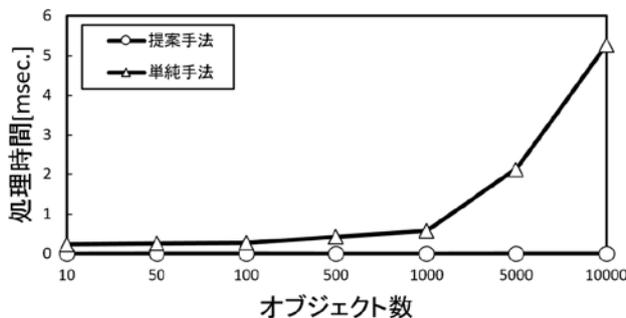


図7：処理時間（オブジェクト追加）

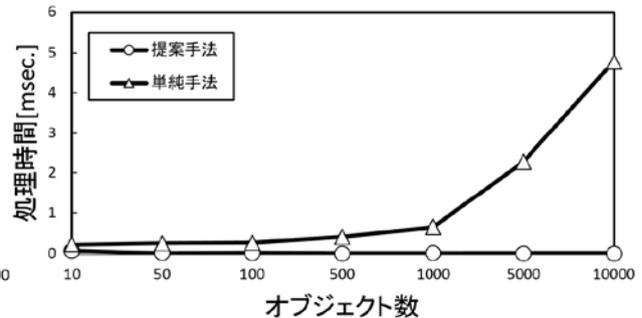


図8：処理時間（オブジェクト削除）

5-4 オブジェクトを削除する場合

オブジェクトを削除する場合におけるRNN領域の更新について、提案手法および単純手法を用いて比較評価を行った。評価では、オブジェクトの削除によってRNN領域の更新にかかる処理時間を100回計測し、平均値を算出した。評価結果を図8に示す。横軸は空間ネットワーク上に存在するオブジェクトの数、縦軸はRNN領域の更新終了までの平均処理時間である。

図8より、オブジェクトの数に関係なく、提案手法の処理時間は単純手法よりも短い。提案手法では、RNN領域の作成に影響しないオブジェクトが削除された場合、垂直二等分線を引かずに処理を終了する。このため、空間ネットワーク上のオブジェクトをランダムに削除する場合、垂直二等分線を引く必要がないオブジェクトを削除する確率は大きくなり、処理速度は0に近づく。

次に、RNN領域の作成に影響するオブジェクトのみを削除した場合、RNN領域の作成に影響するオブジェクトのみを削除した場合においても、追加した場合と同様に、提案手法の処理時間は単純手法に比べて短い。また、提案手法では、RNN領域の作成に影響するオブジェクトを削除する場合、空間ネットワーク上に存在

するオブジェクトを探索する必要がある。このため、オブジェクトを追加する場合と異なり、空間ネットワーク上に存在するオブジェクト数の増加にともない、処理時間は長大化する。

6 おわりに

動的なオブジェクト間の位置関係を考慮した最近傍探索において、オブジェクトの追加および削除に対応してRNN領域を更新する手法を提案した。提案手法では、CZとPRを用いることで、RNN領域を効率よく更新できる。評価の結果、オブジェクトの追加および削除の場合において、どちらも既存手法と比べて処理時間を短縮できることが分かった。例えば、空間ネットワーク上に存在するオブジェクトの数が10000で、RNN領域の作成に影響するオブジェクトのみを削除する場合、提案手法の平均処理時間は2.981ミリ秒、および既存手法の平均処理時間は4.994ミリ秒となり、提案手法は既存手法と比べて平均処理時間を約40.3%短縮できる。

今後は、 k が2以上の場合に対応したアルゴリズムの提案を行う予定である。

【参考文献】

- [1] F. Korn, and S. Muthukrishnan: Influence Sets Based on Reverse Nearest Neighbor Queries, Proceedings of 2000 ACM SIGMOD International Conference on Management of Data, Vol.29, Issue 2, pp.201-212 (2000).
- [2] A. Guttman: R-trees: A Dynamic Index Structure for Spatial Searching, Proceedings of 1984 ACM SIGMOD International Conference on Management of Data, Vol.14, Issue 2, pp.47-57 (1984).
- [3] M. A. Cheema, X. Lin, W. Zhang, and Y. Zhang: Influence zone: Efficiently Processing Reverse k Nearest Neighbors Queries, Proceedings of IEEE 27th International Conference on Data Engineering (ICDE), pp.577-588 (2011).
- [4] K. M. Adhinugraha, D. Taniar, and M. Indrawan: Finding Reverse Nearest Neighbors by Region, Concurrency and Computation: Practice and Experience, Vol.26, Issue 5, pp.1142-1156 (2014).
- [5] W. Wu, F. Yang, C. Y. Chan, and K. L. Tan: FINCH: Evaluating Reverse k-Nearest-Neighbor Queries on Locational Data, Proceedings of VLDB Endowment, Vol.1, Issue 1, pp.1056-1067 (2008).

〈発表資料〉

題名	掲載誌・学会名等	発表年月
A Method to Reduce Processing Time by Parallelizing Generation of Voronoi Diagrams	Proc. the 16th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2018)	2018年11月
A Parallelizing Method for Generation of Voronoi Diagram Using Contact Zone	Journal of Data Intelligence (JDI)	2019年(採録決定済)