

移動体の連続モニタリングのための空間索引構造 及び近接検索方式

渡辺 拓也 Htoo Htoo 大沢 裕

埼玉大学大学院理工学研究科

1 はじめに

位置に関連した情報サービスやカーナビゲーションシステムでの利用を目的として、様々な経路探索が研究されている。この分野では、近年、スマートフォンやタブレットの普及や進化によって、地図やナビゲーションシステムなどを手軽に利用することができるようになり、私達の生活にとってより身近になった。

これらの様々な経路探索において、ユーザや検索対象が固定されているスナップショット検索と、ユーザや対象が移動する連続検索というものが研究されている。連続検索の実生活での応用としては、徒歩などで移動しているユーザに対するタクシーの配車や配送トラック同士での荷物の効率的な受け渡しなどに活用できると考えられる。

しかし一方、この移動体を管理する空間索引構造は更新頻度が高いため高コストになってしまう。そのため、移動体モニタリングに適した空間索引構造が必要となる。また、連続して検索を行う際には移動体の更新時間と検索時間のバランスを取る必要がある。本稿では移動体モニタリングに適した空間索引構造と、それを活用した近接検索方式について提案する。

2 関連研究

移動体の連続近接検索として safe-region という考え方を採用したものが多く研究されてきた [1][2][3]。これらと連続モニタリングを比較した場合、前者のほうが通信コストが下がるものの、生成時の計算量が多いこと、検索点と目的点の双方が移動する場合は生成が難しいという欠点があった。そのため、本稿では連続モニタリング時に効率よく連続近接検索を行うための空間索引構造を提案する。

3 提案方式

本稿では連続検索を効果的に行うために連続モニタリングを活用する。移動体の連続モニタリングに対して R-tree 構造、Quad-tree 構造、Grid 構造の 3 つを適用し、それらの構造で更新時間と連続近接検索の提案

を行い、比較する。近接検索として k-近傍 (kNN) 検索、距離範囲検索の 2 つの実装を行った。道路網距離の計算は A*法 [4] を用いた。

3.1 R-tree 構造

R-tree 構造は最小外接矩形 (MBR) で空間を分割する木構造である。特徴としては各種近接検索が高速に実行できることがあげられる。しかしデメリットとして移動体の更新を行う際に MBR の更新が必要となってしまうため更新時のコストが大きいことがあげられる。

kNN 検索、範囲検索アルゴリズムともに検索点と MBR のユークリッド距離を優先度付きキューを用いて最適優先探索を行い候補点を取得し、それらを道路網距離で検証することによって実装した。

3.2 Grid 構造

Grid 構造は地図平面を縦横等間隔に分割し、移動体を、それが位置するセル内に保存する方法である。特徴としては更新時間のコストが低いということがあげられる。しかしデメリットとして kNN 検索を行う際に検索範囲が広がってしまうため検索に時間がかかることがあげられる。

範囲検索アルゴリズムは IER のようにまずユークリッド距離で検索候補となるセルを探し、その後セル内のオブジェクトを道路網距離で検証する方法で実装した。kNN 検索アルゴリズムは候補点が k 個発見できるまで同心円状に逐次的に検索範囲を広げつつ、k 番目の候補の道路網距離を上限に範囲検索を行い他の候補点を抽出し、道路網距離で更に短いものがないかを確かめる方法で実装した。

3.3 Quad-tree 構造

Quad-tree 構造はノード内のオブジェクトが閾値を超えた場合に 4 つの子ノードを作成することで領域を分割する領域分割型 4 分木で作成した。特徴としては更新時間のコストが高くない上に各種近接検索に対応できるということがあげられる。

kNN 検索、範囲検索アルゴリズムともに R-tree 構造と同様に検索点とノード間のユークリッド距離を対

象とした最適優先探索を行い、候補点を見つけた後に道路網距離での検証を行うことで実装した。

4 性能評価

提案した構造を評価するための実験を行った。アルゴリズムは Java 言語で実装した。また使用計算機は、CPU が Intel Core i7 4790(3.60GHz)、メモリが 16GB のものを用いた。地図は 16,284 ノード、24,914 リンクで構成された実際の道路網を使用した。Grid 構造は 100×100 に分割を行い、Quad-tree 構造は閾値 3 のものと 10 のもので比較した。

まず最初に空間索引構造の更新時間について実験を行った。移動オブジェクトをノードの密度の 0.005 倍から 0.05 倍まで増加させ、一度あたりの更新時間を図示した。図 1 から、Grid 構造と Quad-tree 構造の 2 つが高速に移動体の更新を行えることがわかる。これは、R-tree 構造の MBR 更新に非常に時間がかかるためと考えられる。

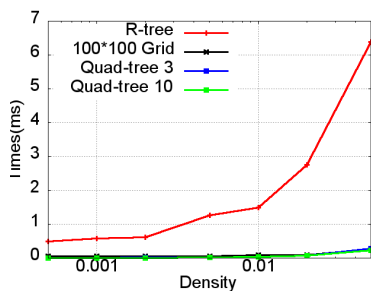


図 1: 空間索引構造の更新

次に kNN 検索の実験を行った。図 2 は $k=3$ とし、移動体の密度を変化させたものであり、図 3 は移動体の密度を 0.01 で固定し k を変化させた。この結果から、kNN 検索は R-tree 構造と Quad-tree 構造が適していると考えられる。Grid 構造は、検索範囲が大きく広がるため他の構造と比べ検索時間が増大すると考えられる。

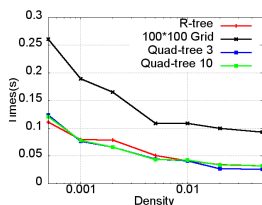


図 2: 密度を変化させたグラフ

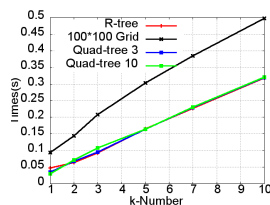


図 3: k を変化させたグラフ

最後に距離範囲検索での実験を行った。図 4 は検索範囲を 3km に固定し、移動体の密度を変化させたもの

である。図 5 は移動体の密度を 0.01 で固定し検索範囲を変化させたものである。今回作成したすべての空間索引構造でほぼおなじ検索時間で処理できていることがわかる。

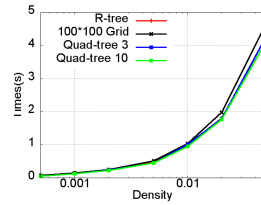


図 4: 密度の変化

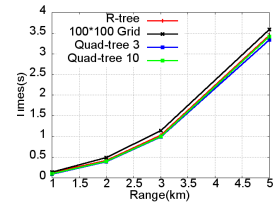


図 5: 検索範囲の変化

5 結論

本稿では移動体の連続モニタリングを効率よく行うための空間索引構造及び移動オブジェクト同士の効率的な近接検索方式について提案した。移動体を空間索引構造に更新する際は Grid 構造及び Quad-tree 構造が高速であり、実際に検索する際には Quad-tree 構造及び R-tree 構造が高速であることがわかった。これら 2 点からどちらも優れている Quad-tree 構造が移動体の連続モニタリングに適していると考えられる。また、今回の実験では Quad-tree の閾値の変化による差があまり見られなかったが、これは移動体の数があまり多くないためと考えられる。今後は移動体数を増やして実験を行い改良したい。

参考文献

- [1] Cheema, Muhammad Aamir, et al. "Continuous monitoring of distance-based range queries." *IEEE Transactions on Knowledge and Data Engineering* 23.8 (2011): 1182-1199.
- [2] Haidar, AL-Khalidi, et al. "On finding safe regions for moving range queries." *Mathematical and Computer Modelling* 58.5 (2013): 1449-1458.
- [3] Cheema, Muhammad Aamir, et al. "Efficiently processing snapshot and continuous reverse k nearest neighbors queries." *The VLDB Journal* 21.5 (2012): 703-728.
- [4] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968): 100-107.