

移動オブジェクトの頻繁な更新に適した空間データ管理構造

A spatial data structure suitable for frequent update of moving objects

西川嘉人 金子裕良 阿部茂
Yoshihito Nishikawa Yasuyoshi Kaneko Shigeru Abe

埼玉大学大学院理工学研究科
Graduate school of Science and Engineering, Saitama University

1. はじめに

移動体通信や GPS の普及に伴い、位置の変化する人や物（以下、移動オブジェクト）の最新位置情報の入手が可能となり、位置情報に基づくサービスが増加している。移動オブジェクトの位置情報管理には R 木や MD 木[1]のような多次元データ管理構造を用いることが多いが、最新位置情報を扱う場合、オブジェクトの継続的な移動による頻繁なデータ更新への対応が問題となる。

我々はオブジェクトの最新位置データのみを管理し、更新処理コストの低減を目的とする。更新では(1)挿入削除による木構造変化、(2)挿入削除時にたどるノード数が主な処理コストとなる。領域分割型である MD 木を拡張し、データ包括型の R 木を拡張した手法と比較を行なう。

2. 移動オブジェクトの更新

移動オブジェクトの更新に関する研究では主に R 木を基にしている[2][3]。R 木は最小包括長方形(MBR: Minimum Bounding Rectangle)でデータを囲い管理するデータ包括型の木構造である。R 木は更新時、MBR の変化に合わせて連鎖的に木構造を変更するため、処理コストが大きい。R 木を改良した手法はいずれも木構造の変化を抑え、更新時の処理低減を図っている。

我々は k-d 木や 4 分木、MD 木のように全体領域を分割してデータを管理する領域分割型の木構造に着目する。領域分割型は領域内のデータ移動に対して木構造の変化は少なく、データ更新における処理コストが小さい。ベクトル管理では検索のために R 木のように MBR を持つ場合もあるが、点データのみを扱う場合は検索のための MBR は無くてもよい。

3. MDF 木(MD-tree for frequent update)

MD 木は領域分割型の木構造で、木の高さはバランスし、他の領域分割型に比べ構築時のメモリコストが小さい。

MDF 木は MD 木を拡張したもので、R 木を用いた手法に比べて以下の特徴がある。(1)MBR を持たず、木構造の変化が小さい、(2)ノード分割が領域 2 分割のため、演算が少なく、(3)管理領域に重なりがほとんどない。

MDF 木は MD 木の削除アルゴリズムを変更し、連鎖的な木構造変化を防いでいる。MD 木の削除アルゴリズムは場合によっては木構造を大きく変化させ処理コストが大きい。この削除の問題は B 木を基にした平衡木の問題で、R 木にも同様に起こる。MDF 木は削除時の木構造変化をリーフ統合のみとし、ノード統合を行なわないことで処理コストを低減した。これにより MD 木の削除時に生じる管理領域の重複も起こらない。

MDF 木では RBU 木の直接リンクを用いた更新法を採用

した。RBU 木の更新は木構造と併せて、オブジェクトとそのオブジェクトが存在するリーフを繋ぐ直接リンクを用いる。これにより削除時に木をたどらず、対象リーフが特定できる。オブジェクトの移動が小さいと仮定すると、対象リーフから root に向け上向きに挿入を行なうことで挿入時にたどるノードを低減できる。

4. 計算機実験

二次元空間(100k×100k)上の移動点オブジェクトを MDF 木、RBU 木で管理し更新時間と検索時間の比較を行った。移動オブジェクトは空間に一樣に分布し、一回の更新による移動距離は $V_m=50$ とした。RBU 木のパラメータ $\epsilon = V_m/10$ 、子ノード数 $M=50$ 、MDF 木のバケット容量 $P=50$ とした。

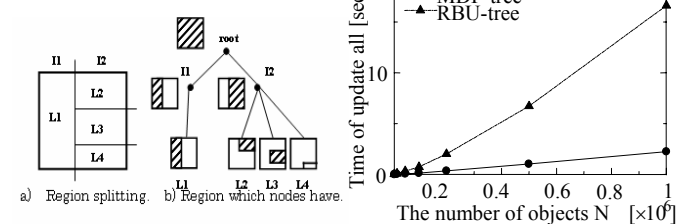


図1 MDF木の領域分割例

図2 更新時間の比較

表1 範囲検索時間 (検索範囲は全体面積を100とした割合)

Search area [%]	0.1	0.2	0.5	1.0	2.0	5.0
MDF-tree [msec]	0.047	0.080	0.32	0.79	1.57	2.49
RBU-tree [msec]	0.092	0.196	0.60	0.87	1.91	2.98

図2にオブジェクト数 N を変化させたときの更新時間を示す。MBR 調整コストや分割コストの大きい RBU 木に比べ、MDF 木は最大 87%更新時間を短縮した。表1に $N=100k$ で検索範囲面積を変化させたときの範囲検索時間を示す。MDF 木は領域に重複がほとんどないため、検索時にたどるノードが少なく、検索がやや高速となる。

5. むすび

移動点オブジェクトの更新処理性能に優れた MDF 木を提案した。データ包括型に比べ、領域分割型は点データの更新管理に適していることを示した。

【参考文献】

- [1]中村, 阿部, "多次元データの平衡木による管理-MD 木-の提案", 信学論(D), J71-D, No9, pp.1745-1752, 1988
- [2]M. L. Lee, W. Hsu, C. S. Jensen, B. Cui and K. L. Teo, "Supporting frequent updates in R-trees: A bottom-up approach", In VLDB, 2003
- [3]X. Xiong and W. G. Aref, "R-trees with Updates Memos", Proc. International Conference on Data Engineering, 22, 2006
- [4]西川, 金子, 阿部"MD木を用いた移動オブジェクト管理" 信学総合大会 D-4-1, 2006