

## 4 分木を用いた移動オブジェクト管理 Management of Moving Objects using Quad-tree

猪田良介 西川嘉人 金子裕良 阿部茂

Ryousuke Inoda Yoshihito Nishikawa Yasuyoshi Kaneko Shigeru Abe

埼玉大学 工学部

Faculty of Engineering, Saitama University

### 1. はじめに

膨大な数の移動オブジェクトの最新位置管理の場合、頻繁なデータ更新に伴う処理時間が問題となる。最新位置管理では R 木を基にしたものが多く発表されている、例えば [1](RBU 木)等である。我々は領域分割形の MD 木[2]に改良を加えた MDF 木を発表した[3]。本稿では、領域分割形で広く知られている 4 分木(Quad-tree)に RBU 木の更新アルゴリズムの一部を付加し、この拡張した 4 分木と MDF 木との比較を行った。データの更新時間に大きな差はなかった。4 分木は初期構築時間で優るが、初期メモリ効率で劣る結果となった。以下に両木の領域分割方法の相違点、長所短所や比較実験結果を示す。

### 2. 4 分木の拡張

4 分木の構成を図 1 (a) に示す。4 分木はリーフにデータを持ち、初期構築時はデータ数がバケット容量  $M$  を超えた場合、領域を単純に 4 等分割していく。

データ更新を高速化するため RBU 木の更新アルゴリズムの一部を採用する。移動オブジェクトの最新データは前データ位置付近に現れるため、同リーフ内となる確率が高い。そこで、データとそのデータを持つリーフとの直接リンクを用い、更新はデータの書き換えのみとする。リーフ外に出る場合は再帰的に親ノードを調べ、対応ノードより挿入を行う。根から行う更新と比べ処理コストの低減になる。MDF 木もこの拡張を行っている。

### 3. 4 分木と MDF 木の特徴

MDF 木は初期構築時、木の高さがバランスするように領域分割を行う。データのソーティングや領域分割のための演算が必要となり、単純 4 等分割の 4 分木に比べ初期構築時間がかかる。

データが一箇所に集中した場合、4 分木は領域を等分割するため木が深くなり、管理領域が分割不可能になる問題があるため、バケット容量  $M$  を大きくとる必要がある。

MDF 木は領域の分割位置が可変のため、この分割の問題は生じない。

MDF 木は挿入時に兄弟リーフ間でデータの調整を行うため、初期構築時のメモリ効率が低い。しかし、データが一箇所に集中した場合、データ削除時の内部ノードの統合を行わないため、頻繁な更新に伴い木が肥大化し、メモリ効率が低下する。4 分木は領域の分割位置が固定のため、初期メモリ効率は高くないが、データ削除時にリーフ統合を行うのでメモリ効率の劣化が少ない。

### 4. 計算機実験

#### 4. 1 実験の設定

二次元空間(100k×100k)上の移動オブジェクトを 4 分木、MDF 木で管理した場合の比較を行う。移動オブジェクトは管理領域内を出ないものとし、一定時間毎にデータが入手可能とした。毎回の移動距離は管理領域の 0.05%以下として乱数で発生させた。バケット容量

$M=50$  とし、移動オブジェクト数  $N=100k$  とした。データは領域全体に様に分布させた一様分布形と、半数のデータを一様、残りを中心 1k 四方に発生させた局所集中形の 2 種類のデータ分布を用意した。初期構築時間、更新時間、初期メモリ効率、100 回更新後のメモリ効率について比較を行った。局所集中形で 1k 四方の検索範囲が与えられた場合、集中領域の検索時間について比較を行った。

**4. 2 実験結果** 木の比較実験結果を表 1 に示す。初期構築時間は 4 分木が約 1/4 と短い。これは 4 分木の単純分割に起因している。初期メモリ効率は MDF 木が構造上高いが、継続的な更新に伴い低下する。4 分木はリーフの統合が起こるためメモリ効率の悪化はない。更新時間は更新時のアルゴリズムが同様のため、ほぼ同じである。

局所集中形の場合、木がバランスする構造の MDF 木は経由ノードが少ないため、検索時間が短い。

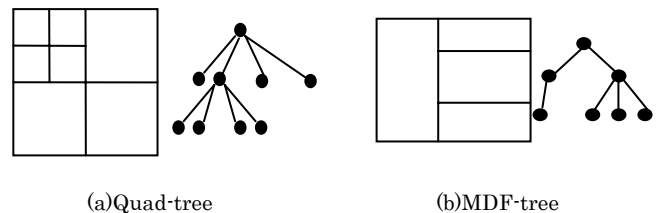


図 1. 木の構成

表 1 4 分木と MDF 木の比較(データ数 100k)

データ分布	一様分布形		局所集中形	
	4 分木	MDF 木	4 分木	MDF 木
初期構築時間[sec]	0.10	0.40	0.11	0.39
更新時間[sec]	0.055	0.044	0.17	0.17
メモリ効率(初期)[%]	48.8	81.5	37.8	78.0
(更新 100 回後)[%]	48.8	78.4	44.4	48.6
検索時間[msec]			6.78	5.49

### 5. むすび

4 分木と MDF 木の比較検討を行った。両木のデータ更新時間に大きな差はない。4 分木は初期構築時間が短い。また初期のメモリ効率は悪いが、リーフの統合が容易に行えるためメモリ効率の劣化が少ない。MDF 木は検索時間が短く、データが局所集中しても分割限界がないという長所がある。

【参考文献】

- [1] 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
- [2] 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)
- [3] 西川, 金子, 阿部 "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" 全国大会 (2006 年度)