

移動オブジェクトを対象とした 時空間データ管理手法とその評価

王 軼 群[†] 野 澤 博[†] 土 方 嘉 徳[†]
仲 谷 美 江[†] 西 田 正 吾[†]

本稿では、移動オブジェクトを対象とした時空間データの高速検索手法である XAT (eXtended Adaptive Tree) 構造を提案する。XAT 構造による検索は、最初に検索範囲に入りそうなオブジェクトを空間木と時間木から構成される木構造で絞込み、その後各オブジェクトの移動線分データが検索範囲に入るか否かのチェックを行う。計算機実験により、XAT 構造と従来方式の 3D 管理構造と比較をした。その結果、任意の検索範囲において、XAT 構造は 3D 管理構造より優れていることが分かった。

Spatio-Temporal Data Management Method for Moving Objects and Its Evaluation

YIQUN WANG,[†] HIROSHI NOZAWA,[†] YOSHINORI HIJIKATA,[†]
MIE NAKATANI[†] and SHOGO NISHIDA[†]

This paper deals with spatio-temporal indexing method for moving objects. In our research, we propose XAT (eXtended Adaptive Tree) structure, consisting of spatial trees and temporal trees, for fast search for spatio-temporal data. The searching process in XAT structure is divided into two steps. The first step roughly narrows down the potential solutions (moving objects) according to the given searching range. The last step fixes the real solution by checking the object's moving track. We compare XAT structure and 3D structure, one of the conventional methods, by computer simulation. The result shows that XAT structure works faster than 3D structure when there is difference between the spatial search range and temporal search range.

1. はじめに

携帯電話上での商用サービスや、GPS と GIS を組み合わせた防災システムなど、移動端末を利用したアプリケーションが増えつつある¹⁾。これらのアプリケーションにおいては、ある時刻にある場所に存在した端末のデータを参照したり、逆にその端末にデータを送信したりする必要がある。このためには、移動端末（一般的には移動オブジェクト）を時空間データ（位置と時間情報で構成されるデータ）に基づき、効率的に管理する必要がある。

筆者らは静止オブジェクトを対象として、ある検索範囲（空間範囲と時間範囲）中に含まれるオブジェクトを効率的に検索する方式である AT 構造 (Adaptive Tree 構造) を提案してきた²⁾。一般的にデータを木構

造で管理することは高速検索の基礎となるが、検索範囲が広がると多くのノードや葉ノードを探索することになり、検索速度が低下する問題がある。AT 構造は、空間データから作成した木構造（空間木）と、時間データから作成した木構造（時間木）を用意しておき、検索範囲から、空間の検索範囲と時間の検索範囲の大小を比較し、検索範囲の狭い方の木構造を選択する方式である。

本研究では、AT 構造の考え方を移動オブジェクトの検索に拡張した XAT 構造 (eXtended Adaptive Tree 構造) を提案する。一般的に、ある一つの移動オブジェクトの時空間データは、その移動軌跡を一定の時間間隔で区切った移動線分データとして表現される^{3)~6)}。最も基本的な検索方式としては、この移動線分データを直接木構造で管理する方式が考えられる⁵⁾。しかし、この方式では、移動オブジェクトの数が増大した時に検索時間が増大するだけでなく、より動きを詳細に追跡しようとした時にも、移動線分データの数が増大し、

[†] 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

検索時間が増大する問題がある。

XAT 構造ではこの問題に対し、最初に検索範囲に入りそうなオブジェクトを空間木と時間木から構成される木構造で絞り込み、その後各オブジェクトの移動線分データが検索範囲に入るか否かのチェックを行なうことで検索時間の高速化を図る。

本稿では、2 節で関連研究について述べる。3 節で、筆者らが提案する XAT 構造について説明する。4 節では、XAT 構造を計算機実験により評価する。最後に、5 節でまとめを述べる。

2. 関連研究

時空間データの管理方式には、様々な方法論が提案されている。これらの多くは、多次元データを木構造で管理するための基礎的なアルゴリズムである **k-d 木**⁷⁾、**R-tree**⁸⁾、**BD 木**⁹⁾、**MD 木**¹⁰⁾ などを応用している。どのような観点から検索速度の向上を目指すかと言う点に着目すると、大きくは次の 3 種類に分類される。

- (1) 時間に対して、検索速度向上を目指す方式
MT(Multi-Tree) 構造とその発展型 (**MR-Tree**¹¹⁾、**MP Tree**¹²⁾、**HR-Tree**¹³⁾、**Overlapping Linear Quad-Trees**¹⁴⁾、**2+3 R-tree**⁵⁾
- (2) 空間に対して、検索速度向上を目指す方式
ST(Single-Tree) 構造とその発展型 (**RT-Tree**)¹⁵⁾
- (3) 時間と空間対して、均一の検索速度を目指す方式
3D(3 Dimensional) 管理構造とその発展型 (**3D R-Tree**)¹⁶⁾

MT 構造とその発展型は、空間情報のみに基づいた木構造をいくつかの時区間ごとに作成する。現在 (最新) の時間のデータに対して検索を行なうことが多い場合や、過去の検索においても、検索条件の時間範囲が狭い場合に有効な手法である。

ST 構造とその発展型は、空間情報にのみ基づいた木構造を一つだけ作成し、時間情報は葉ノード中にあるオブジェクトのデータの属性として扱う。検索条件において、空間範囲が狭い場合に有効な手法である。

3D 管理構造とその発展型は、位置 (2 次元) と時間 (1 次元) で構成される 3 次元空間を木構造で管理するものである。木構造は、位置 (x,y) と時間 (t) の 3 つの属性を繰り返し順番に用いることで作成される (実用上、高さに相当する z 軸に関するデータは、**DEM** と呼ばれるオブジェクトの属性値として扱われることが多い¹⁷⁾)。時間と空間、また現在と過去に対して、均一の検索速度を提供するため、分類した 3 種類の手

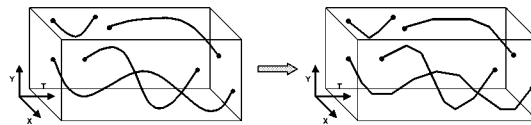


図 1 移動物体の移動軌跡の線分化
Fig. 1 Line segments of moving objects.

法の中では、最も汎用性の高い手法である。

筆者らが提案する XAT 構造は、3D 管理構造と同じく、時間と空間に対して均一の検索速度を提供することを目指すものである。XAT 構造が 3D 管理構造と異なるのは、(1) 1 つの木構造で時空間データを管理するのではなく、時間データを用いる木構造と空間データを用いる木構造を別々に作成して適応的に切り替える点と、(2) 一度オブジェクト単位で検索候補を絞り込んでから詳細な移動経路をチェックする点にある。本稿では、4 節にて XAT 構造と 3D 管理構造をシミュレーション実験により比較している。

3. XAT 構造

3.1 移動オブジェクトデータ

移動オブジェクトとは、時間の経過と共に空間中を移動するオブジェクトのことである (以降、特に必要がない限り単にオブジェクトと呼ぶ)。その移動経路は曲線として表されるが、この曲線を計算機で管理することは難しい。そこで、一般的にオブジェクトの移動経路データの管理においては、移動経路を微小時間で区切ることによって複数の線分のつながりとして考え、その線分群を管理する (図 1)^{3)~6)}。本研究においても、オブジェクトのデータを、この移動線分で表すものとする。

3.2 AT 構造と XAT 構造

AT 構造における基本原理は、データを管理する木構造を、時間情報のみから作成するもの (時間木) と、空間情報のみから作成するもの (空間木) に分けて構築しておき、検索条件の時間範囲 (以下、時間検索範囲) と空間範囲 (以下、空間検索範囲) の割合の大きさから、検索条件の厳しい方の木構造を検索することにある。²⁾ XAT 構造は、この考え方を移動オブジェクトの検索にも適用できるように拡張したものである。

XAT 構造では、検索の高速化のために、時間木または空間木から検索結果の候補となるオブジェクトを絞り込んだ後、そのオブジェクトの各移動線分が検索範囲に含まれるか否かを検査している。さらに、検索の高速化のために、オブジェクト絞り込み時と、移動線分データチェック時に、次のような工夫を行なう。

- (1) オブジェクト絞り込みの高速化

本来時間軸に線分(区間)データとして表されるオブジェクトの存在時間範囲の情報を、点データに射影する。

(2) 移動線分データチェックの高速化

移動線分データを先頭から逐次的にチェックするのではなく、これらも空間木と時間木で管理する。

木構造で管理する対象データが線分データであれば、各ノードの分岐条件は範囲となる。すると、異なるノードであっても、その分岐条件の範囲には重なりができることがある。これが点データであれば、各ノードの分岐条件は閾値となり、分岐条件に重なりのあるノードはできない。この違いは、検索するノード・葉ノード数に影響するため、検索時間にも影響する。前者の工夫は、この問題に着目したものである。また、後者の工夫は、移動軌跡を詳細に追跡しようとした時に、移動線分データ数が増加し、検索時間も増加するという問題に着目したものである。

XAT 構造における処理の流れは図 2 に示され、大きくはオブジェクトを管理する部分(以下、オブジェクト管理部)と、各オブジェクトごとにその移動線分を管理する部分(以下、線分管理部)に分けられる。オブジェクト管理部、線分管理部はそれぞれ、検索判断ロジック部、空間木、時間木の 3 つの部分から構成される。XAT 構造の線分管理部の処理の流れはオブジェクト管理部のそれとほぼ同じである。唯一の違いは、オブジェクト管理部の管理対象がオブジェクトであるのに対し、線分管理部の管理対象は特定のオブジェクトの移動線分であることにある。以下の節では、XAT 構造のオブジェクト管理部についてだけその詳細を説明する。

3.3 オブジェクト管理部

3.3.1 検索判断ロジック部

オブジェクト管理部における検索判断ロジック部では、全時間範囲に対する時間検索範囲の割合と、全空間範囲に対する空間検索範囲の割合を求め、その割合の小さい方の木構造を選択する。

3.3.2 空間木

オブジェクト管理部における空間木は、オブジェクトの 3 次元空間(位置+時間)中における移動経路情報のうち、位置に関する情報のみを使い、オブジェクトを管理する木構造である。3 次元空間中の移動線分群から、時間軸を無視して空間平面上に射影してできた線分群をオブジェクト単位でまとめ(図 3)、その中心点を k-d 木を用いて管理する。k-d 木とは、最も基礎的な多次元空間データの管理手法で、多次元空間中

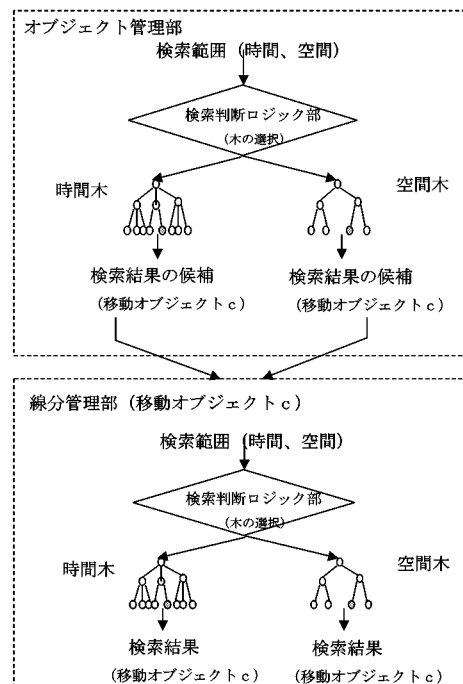


図 2 XAT 検索の流れ
Fig. 2 Flow of search in XAT.

の点データ群を、データの個数が半分ずつになるように空間を分割して、それを木構造にしたものである⁷⁾。

木構造の中間ノードにおける分岐条件の設定のために、空間平面上に射影されたオブジェクトの移動軌跡に外接長方形(MBR: minimum bounding rectangle)¹⁷⁾を設定する(図 4)。外接長方形は、オブジェクトのすべての線分を含み、なおかつ最小になるように設定する。バケット型管理方式を用いることとし、葉ノード中の全オブジェクトの外接長方形を含む最小の長方形をその葉ノードの外接長方形とする。各中間ノードにおいては、その 2 つの子ノードの外接長方形を共に含む最小の長方形をそのノードの外接長方形とする。以上の手順を葉ノードから根ノードに向かって再帰的に行なうことにより、全てのノードに分岐条件として外接長方形の情報を付加する。

空間木を検索する方法は、空間検索範囲と現時点のノード P に付加した外接長方形を比較し、重なりがある子ノードをたどっていく。子ノードに重なりがある限り再帰的に領域の比較を繰り返し、到達したすべての葉ノードにあるオブジェクトを獲得する。獲得したオブジェクトにおいて、その存在時間範囲と時間検索範囲とに重なりがあるか否かを検査し、重なりがあればそのオブジェクトを検索結果の候補として出力する(これは、不要にオブジェクトの木構造を探索する

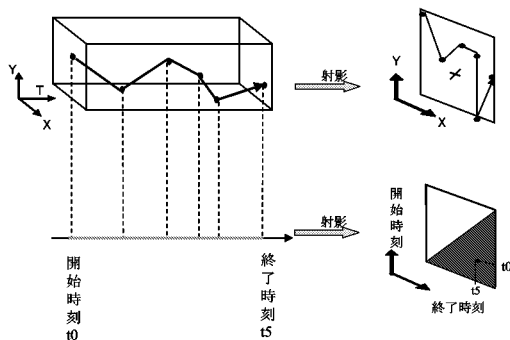


図 3 空間平面、時間軸への射影
Fig. 3 Projection on space plane and time axis.

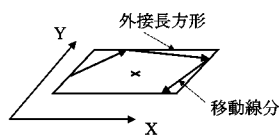


図 4 外接長方形

ことを防ぐためである)。

3.3.3 時間木

オブジェクト管理部における時間木は、オブジェクトの 3 次元空間 (位置+時間) 中における移動経路情報のうち、時間に関する情報のみを使い、オブジェクトを管理する木構造である。ここではまず、オブジェクトの開始 (発生)・終了 (消滅) 時間を $X \cdot Y$ 軸にとり、図 3 のように 2 次元平面上の点情報に変換する。このことにより時区間データを点情報として、データ構造化することが出来る。このようにデータ変換を行うと、全てのデータは図 5 に示すように領域 $y \leq x$ ($x \geq 0, y \geq 0$) の三角形の範囲に含まれることになる。なぜなら、開始時間は終了時間より必ず前にあるからである。この三角形の範囲内にある点情報は 3 分木を用いて管理される (図 5)。

時間木の作り方は、まずデータを終了時間でソーティングし、終了時間の小さい方から全体の 4 分の 1 にあたるデータを探し出す。次に、データの開始時間に着目し、開始時間の小さい方から全体データ数の 4 分の 1 にあたるデータを探し出す。その後、この 2 つのデータの平均値 a を用いて時間データ分布を図 5 のように 3 分割する。三角形になった領域は以下再帰的に分割していく。長方形領域は $k-d$ 木を用いて 2 分木で再帰的に分割を繰り返す。

時間木を検索する方法は、時間検索範囲と現時点のノード P における分岐条件を比較し、重なりがある子ノードをたどっていく。子ノードに重なりがある限り再帰的に領域の比較を繰り返し、到達したすべての

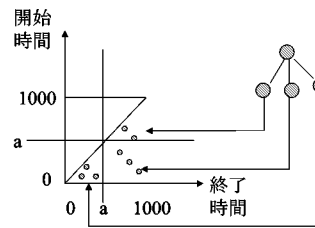


図 5 時間木の構築方法
Fig. 5 Construction of temporal tree.

葉ノードにあるオブジェクトを獲得する。獲得したオブジェクトにおいて、その移動範囲と空間検索範囲とに重なりがあるか否かを検査し、重なりがあればそのオブジェクトを検索結果の候補として出力する (これは、不要にオブジェクトの木構造を探索することを防ぐためである)。

4. 計算機実験

4.1 評価の目的

本節では、計算機上のシミュレーション実験により XAT 構造の評価を行う。本評価の目的は、(1) 最も代表的な既存の時空間データ管理手法である 3D 管理構造と比較を行うことと、(2) XAT 構造で我々が提案した高速化のための工夫が検索時間の短縮に貢献しているのか否かを明らかにすることにある。3 節でも述べたように、XAT 構造における高速化の工夫点は以下の 3 点にある。

(1) オブジェクト絞り込み

検索結果となる可能性のあるオブジェクトを絞り込んでから、詳細な動きを検査する。

(2) 木構造の二重化

移動線分データを先頭から逐次的にチェックするのではなく、これらも空間木と時間木で管理する。

(3) 時間情報の点データ化

本来時間軸に線分データとして表されるオブジェクトの存在時間範囲の情報を、点データに射影する。

本評価では、3D 管理構造を実装し、これらの工夫を行わないバージョンの XAT 構造も、それぞれ XAT 構造 (Type 1)、XAT 構造 (Type 2)、XAT 構造 (Type 3) として実装する (それぞれの手法の詳細は 4.2 節で説明する)。その後、これらとオリジナルの XAT 構造を比較する。

4.2 比較対象手法の詳細

(1) 3D 管理構造

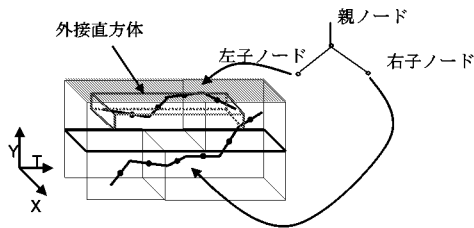


図 6 3D 管理構造
Fig. 6 3D structure.

位置 (2次元) と時間 (1次元) で構成される 3次元空間を木構造で管理するものである。具体的には、移動線分を中心点で代表させ、その中心点を k-d 木を用いて管理する (図 6)。線分の領域情報は、開始点と終了点の 2 点を含み、時間軸と空間軸の 3 つの軸を使って作成できる外接直方体として表現する。木構造のノードでは、この外接直方体のデータを分岐条件として設定している。

(2) XAT 構造 (Type 1)

XAT 構造のようにオブジェクトを絞り込む木構造と移動線分をチェックするための木構造を持つのではなく、1 つの時間木と空間木で直接すべてのオブジェクトの移動線分データを管理する。

(3) XAT 構造 (Type 2)

XAT 構造の線分管理部において時間木と空間木を用いて移動線分データを管理するのではなく、線形のリストで移動線分データを管理する。つまり、オブジェクト管理部において時間木または空間木によりオブジェクトを絞り込んだ後は、各オブジェクトの移動線分データを一つずつチェックする。

(4) XAT 構造 (Type 3)

XAT 構造のように、オブジェクト及び移動線分の存在時間情報を、開始時間軸と終了時間軸から構成される 2次元平面上に点データとしてマッピングするのではなく、そのまま 1次元の時間軸上に表される線分データとして扱う。時間軸上でオブジェクトまたは移動線分の存在時間範囲の中心点の数がちょうど 2 分ずつされるように木構造を作成した後、各ノードに、存在時間の範囲情報を分岐条件として設定している (図 7)。

4.3 標準実験条件

実験用にオブジェクトの移動データを作成した。今回の実験では移動の仕方として、直線運動、ランダム運動、正弦波運動の 3 種類を作成した。これらの運動を組み合わせれば、現実のほとんどの移動データを表現できると考え、この 3 種類を選択した。

本比較では、いくつかの実験を行うが、表 1 に標準

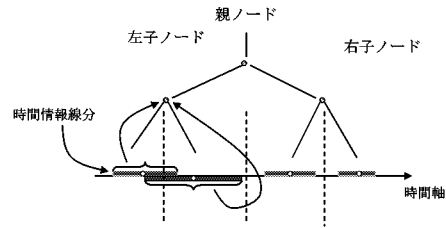


図 7 XAT 構造 (Type 3) における時間木
Fig. 7 Temporal tree in XAT Structure (Type 3).

表 1 標準実験条件

Table 1 Parameters for experiments.

オブジェクト数	1000
全時空間範囲	[0,0,0]-[1000,1000,1000]
オブジェクトの移動範囲	全空間範囲の 1%(面積)
オブジェクトの存在時間範囲	全時間範囲の 20% (200 単位時間)
線分化の細かさ	1 単位時間おき
検索条件	検索条件 A または検索条件 B のどちらか (検索条件 A) 空間:全空間範囲の 10%(面積) 時間:全時間範囲の 5% (検索条件 B) 空間:全空間範囲の 5%(面積) 時間:全時間範囲の 10%

とする実験条件を示す。この後の実験で実験条件の各種パラメータを変更する場合は、この値を中心として変更する。これらのパラメータのうち、オブジェクトの移動範囲は、その左端、右端、上端、下端から構成される長方形の面積で設定している。時間検索範囲と空間検索範囲には、2 通りの組み合わせ (検索条件 A と検索条件 B) を標準値とする。

実験には、Pentium4(1.5GHz) のマシンを使用し、プログラミング言語には C++ を使用した。また、各葉ノードでは、最大 10 個のデータを保管可能とした。

4.4 実験方法

本評価では、データ構造作成の観点から、データ構造作成時間、作成ノード数、作成葉ノード数を用いる。また、データ構造検索の観点から、検索時間を用いる。

まず、データ構造作成の観点から、XAT 構造と 3D 管理構造を比較する。ここでは、標準実験条件において、データ構造を作成する。

次に、データ構造検索の観点から、XAT 構造と 3D 管理構造、XAT 構造 (Type 1)、XAT 構造 (Type 2)、XAT 構造 (Type 3) を比較する。実験条件のパラメータのうち、ここでは以下の 2 つに注目する。

- (1) 時間検索範囲と空間検索範囲
- (2) オブジェクトの移動範囲と存在時間範囲

(1)は、検索という利用面を考えた場合、最も頻繁に変更のありうるパラメータである。本評価では、これらを変更することで、XAT構造と3D管理構造を比較する。また、XAT構造(Type 1)、XAT構造(Type 2)、XAT構造(Type 3)とも比較し、3つの高速化の工夫が、検索時間の向上に貢献しているのか否かを検証する。(2)は、オブジェクトが持つ特性である。これらを変更して、一度オブジェクト単位で管理するXAT構造と、直接移動線分を管理する3D管理構造を比較し、XAT構造がいかなる条件でも優れた検索速度を実現するのか否かを検証する。

4.5 実験結果

4.5.1 データ構造作成

表2に、XAT構造及び3D管理構造のデータ構造作成時間、ノード数、葉ノード数を示す。作成時間に注目すると、3D管理構造よりXAT構造の方が短いことが分かる。これは、k-d木の作成におけるソーティングの影響である。k-d木では、データ数が半分ずつになるように各軸に平行な線で分割するため、データをソートしておく必要がある。3D管理構造では、すべてのオブジェクトの移動線分を一度にソートする必要があるが、XAT構造では移動線分をオブジェクトごとにソートすればよい。そのため、XAT構造ではソーティングにかかる時間が短くなっている。

ノード数、葉ノード数に注目すると、XAT構造より3D管理構造の方が少ないことが分かる。これは、XAT構造では時間木と空間木の両方を作成する必要があるためである。

これらのことから、XAT構造はデータ構造作成時間は早い、メモリコストは悪くなる、ことが分かる。

4.5.2 データ構造検索

(1) 時間検索範囲と空間検索範囲

図8-(a)に、時間検索範囲を10%に固定し、空間検索範囲を1%-5%-10%-15%-20%-40%-60%と変化させたときの検索時間を示す。図8-(b)に、空間検索範囲を10%に固定し、時間検索範囲を1%-5%-10%-15%-20%-40%-60%と変化させたときの検索時間を示す。ここでは、直線運動の結果のみ示しているが、ランダム運動や正弦波運動についても、ほぼ同様の結果が得られている。

時間検索範囲(空間検索範囲)に対し、空間検索範囲(時間検索範囲)が大きくなるにつれて、XAT構造と3D管理構造の検索時間の差は大きくなる傾向がある。このことから、特に時間と空間の検索範囲の偏りが大きくなる時に、XAT構造は3D管理構造より

も優れた検索性能を発揮することが検証された。

また、XAT構造と高速化の工夫を行わないXAT構造(Type 1~3)を比較しても、XAT構造はいずれの方式よりも、検索時間が短いことが分かる。このことから、我々の提案する3つの高速化の工夫が、実際に検索時間の短縮に貢献していることが検証された。

(2) オブジェクトの移動範囲と存在時間範囲

図9-(a),(b)に、オブジェクトの移動範囲を、全空間範囲の0.1%-0.5%-1%-5%-10%-20%と変化させたときのグラフを示す。図9-(c),(d)に、オブジェクトの存在時間範囲を、50-100-200-400-600時間単位(全時間範囲の5%-10%-20%-40%-60%)と変化させたときのグラフを示す。(a),(c)は時間検索範囲5%で空間検索範囲が10%、(b),(d)は時間検索範囲10%で空間検索範囲が5%である。ここでは、直線運動の結果のみ示しているが、ランダム運動や正弦波運動についても、ほぼ同様の結果が得られている。

図9-(a),(b)から、オブジェクトの移動範囲が広い場合は、XAT構造は3D管理構造に比べて、検索時間の悪化が急であることが分かる。これは、XAT構造が一度オブジェクト単位で絞り込みを行っているためである。全空間に対してオブジェクトの存在する割合が高くなると、検索範囲が同じでも、その範囲に入るオブジェクトの数が多くなる。すると、XAT構造では移動線分までチェックする必要のあるオブジェクトの数が増えるため、その検索時間が悪化するのである。

図9-(c),(d)から、オブジェクトの存在時間範囲が長い場合は、XAT構造と3D管理構造の検索時間の変化には大きな差は見られない。XAT構造では、全時間に対するオブジェクトの存在する割合が高くなると、移動線分までチェックする必要のあるオブジェクトの数が増えるため、検索時間が悪化する。それに対し、3D管理構造でも、オブジェクトの存在時間が長くなることで、1つのオブジェクト当たりの移動線分数が増えるため、検索時間が悪化する。XAT構造と3D管理構造の双方で、検索時間が悪化する要因があるため、検索時間の変化にも大きな差は見られなかったと言える。

これらのことから、オブジェクトの存在時間が長くなっても、XAT構造における検索時間の悪化の程度は、3D管理構造のそれと大きな差はないことが分かった。しかし、オブジェクトの移動範囲が大きくなった場合は、XAT構造は3D管理構造よりも検索時間が悪化することが分かった。

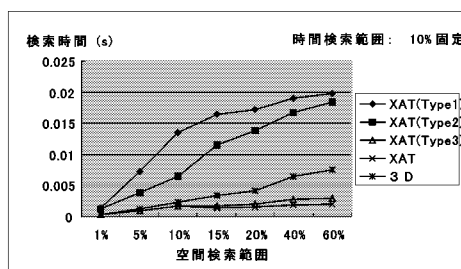
4.6 結論

XAT構造では、オブジェクトの詳細な動きまで追

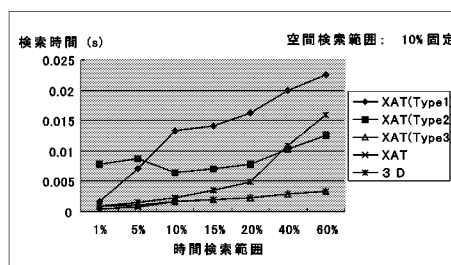
表 2 データ構造作成コスト

Table 2 Cost for data structure creation.

	3D 管理構造	XAT 構造	XAT 構造 (内訳)			
			オブジェクト管理部		線分管理部	
			時間木	空間木	時間木	空間木
作成時間 (s)	10.71	3.58	0.01	0.02	1.30	2.07
ノード数	32767	57260	127	133	26006	30994
葉ノード数	32768	85271	128	143	53015	31985

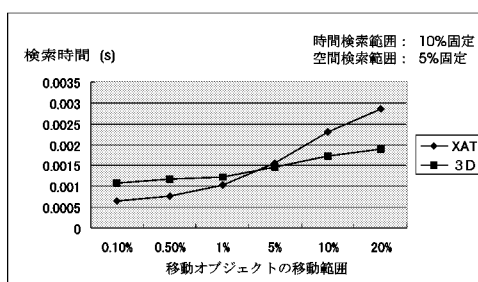


(a) 直線運動

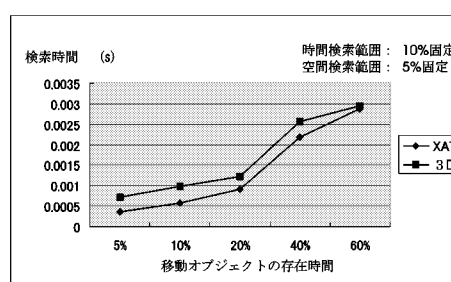


(b) 直線運動

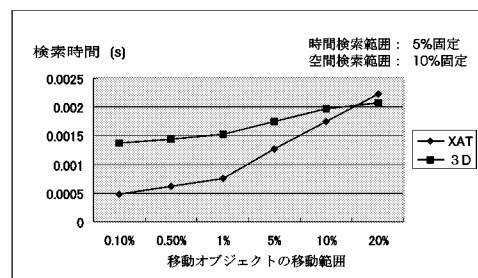
図 8 検索範囲に対する検索時間
 Fig. 8 Search time for search range.



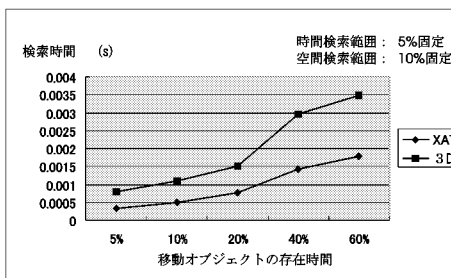
(a) 時間検索範囲: 10%
 空間検索範囲: 5%



(c) 時間検索範囲: 10%
 空間検索範囲: 5%



(b) 時間検索範囲: 5%
 空間検索範囲: 10%



(d) 時間検索範囲: 5%
 空間検索範囲: 10%

図 9 移動範囲/存在時間範囲に対する検索時間
 Fig. 9 Search time for moving range and existing time.

跡した場合の時空間データに対しても、高速に検索できることを目指した。XAT 構造には、メモリコストがかかる欠点と、オブジェクトの移動範囲が広い場合に検索時間が長くなる欠点は見られたが、任意の検索範囲において、XAT 構造は 3D 管理構造よりも優れ

ていることが分かった。

今後、GPS 方式や基地局による三角測量方式¹⁸⁾、セル ID 方式¹⁸⁾などの、人や端末の位置を正確に測定する技術は、ますます向上してくるものと考えられる。これに伴い、詳細な移動データを用いたアプリケー

ション (タウン情報, 動体管理, 交通情報など) も増加してくるものと考えられる¹⁸⁾。このようなアプリケーションにおいては, XAT 構造のような特性を持つ検索方式は有効になってくるものと考えられる。

5. ま と め

本稿では, 移動オブジェクトを対象とした時空間データを高速に検索する手法を提案した。本手法では, オブジェクトを管理する時間木及び空間木と, 移動線分を管理する時間木及び空間木を用意しておき, (1) 検索範囲に応じて時間木と空間木を適応的に切り替える, (2) 一度オブジェクト単位で検索結果の候補を絞り込んでから詳細な動きを検査する, ことを行っている。従来方式である 3D 管理方式とシミュレーション実験により比較した。その結果, 提案する手法は, メモリコストがかかる欠点と, オブジェクトの移動範囲が広い場合に, 検索時間が長くなる欠点が見られたが, 細かい動きを有するデータである場合は, 3D 管理構造より優れていることが確認できた。

今後は, 逐次データが投入・削除されるような動的データを扱えるように, 提案した手法を拡張していきたいと考えている。

謝 辞

本研究は, 文部科学省科学研究費 (No. 13S0018, No. 12480097) の助成を受けて行われた。

参 考 文 献

- 1) 高橋克巳, 寺岡文男: 位置情報を利用したモバイルコンピューティング特集, 情報処理学会学会誌, Vol. 42, No. 4, pp. 353-369, (2001).
- 2) 池本和生, 野澤博, 仲篤起, 才脇直樹, 西田正吾: 時空間ウォークスルーのためのデータ管理の一方式, 電気学会論文誌, Vol. 121-C, No. 1, pp. 142-149, (2001).
- 3) M. Nabil, A.H.H. Nuu and J. Shepherd: Modelling Moving Objects in Multimedia Databases, Proc. of the 5th International Conference on Database Systems for Advanced Application, pp. 67-75, (1996).
- 4) M. Erwig, et al.: Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases, GeoInformatica, Vol. 3, No. 3, pp. 269-295, (1996).
- 5) M.A. Nascimento, J.R.O. Silva and Y. Theodoridis: Evaluation of Access Structures for Discretely Moving Points, Proc. of the Intl. Workshop on Spatiotemporal Database Management (STDBM'99), pp. 171-188, (1999).

- 6) 鶴飼規子, 増永良文: 時空間データベース構築のためのムービングオブジェクトモデル, 情報処理学会研究報告, 99-DBS-119, pp. 153-158, (1999).
- 7) J.L. Bentley: Multi dimensional Binary Search Trees Used for Associative Searching, Comm. of the ACM, Vol. 18, pp. 509-517, (1975).
- 8) A. Guttman: R-Trees: A Dynamic Index Structure for Spatial Searching, Proc. of ACM SIGMOD Intl. Symp. on the Management of Data, pp. 45-57, (1984).
- 9) 大沢裕, 坂内正夫: 良好な動特性を持つ多次元点データ管理構造の一提案, 電子通信学会論文誌 D, Vol. J66-D, No. 10, pp. 1193-1200, (1983).
- 10) 中村泰明, 阿部茂, 大沢裕, 坂内正夫: 多次元データの平衡木による管理 - MD 木の提案, 電子通信学会論文誌 D, Vol. J71-D, No. 9, pp. 1745-1752, (1988).
- 11) M.A. Nascimento, J.R.O. Silva, and Y. Theodoridis: Evaluation of Access Structure for Discretely Moving Points, Spatio-temporal Database Management 1999, pp. 171-188, (1999).
- 12) 寺岡昭彦, 丸山稔, 中村泰明, 西田正吾: 空間検索を効率化した時空間データ管理構造の提案 - 多次元 Persistent Tree -, 電子情報通信学会論文誌 D-II, Vol. K78-D-II, No. 9, pp. 1346-1355, (1995).
- 13) M.A. Nascimento and J.R.O. Silva and Y. Theodoridis: Access Structures for Moving Points, Technical Report 33, TimeCenter, (1998).
- 14) T. Tzouramanis, M. Vassilakopoulos and Y. Manolopoulos: Overlapping Linear Quadrees: A Spatio-temporal Access Method, Proc. of the 6th ACM Intl. Workshop on Geographical Information Systems, pp. 1-7, (1998).
- 15) X. Xu, J. Han and W. Lu.: RT-tree: An Improves R-tree Index Structure for Spatiotemporal Databases, Proc. of the 4th Intl. Symposium on Spatial Data Handling, pp. 1040-1049, (1990).
- 16) Y. Theodoridis, M. Vazirgiannis and T. Sellis.: Spatio-temporal Indexing for Large Multimedia Applications, Proc. of the 3rd IEEE Conf. on Multimedia Computing and Systems, pp. 441-448, (1996).
- 17) 大沢裕, 中村泰明: 空間データの効率的な管理と高速空間検索のためのデータ構造, 情報処理学会学会誌, Vol. 42, No. 10, pp. 965-971, (2001).
- 18) 太田洋ほか: 位置情報と携帯電話, 情報処理学会学会誌, Vol. 42, No. 4, pp. 358-361, (2001).