時空間データ管理方式「XAT 構造」の種々の検索要求に対する評価

王　軼群†　土方　嘉徳†　西田　正吾†

† 大阪大学大学院基礎工学研究科

E-mail: †wang@nishilab.sys.es.osaka-u.ac.jp, ††{hijikata,nishida}@sys.es.osaka-u.ac.jp

あらまし　本稿では、我々が提案してきた移動オブジェクトに対する時空間データ管理方式である XAT 構造を、種々の検索要求から評価する。XAT 構造は、木構造を時間木と空間木に分けて、適応的に切り替える方式である。また高速検索のため、解候補の絞込みを行う工夫を行っている。本評価では、この XAT 構造を代表的な時空間データ管理方式である 3D 管理構造と比較する。具体的には、(1) 移動オブジェクトの範囲検索、(2) 移動オブジェクトの線分軌跡の範囲検索、(3) 移動オブジェクトの線分軌跡の最近接検索に対する比較を行う。また、 3D 管理構造に XAT 構造で行った工夫（解候補の絞込み）をした場合との移動オブジェクトの範囲検索において比較する。

キーワード　時空間 DB、範囲検索、最近接検索、移動オブジェクト

Evaluation of the Spatio-temporal Data Structure for Various Kinds of Search Request

Yiqun WANG†, Yoshinori HIJIKATA†, and Shogo NISHIDA†

† Graduate School of Engineering Science, Osaka University

E-mail: †wang@nishilab.sys.es.osaka-u.ac.jp, ††{hijikata,nishida}@sys.es.osaka-u.ac.jp

Abstract　This paper deals with spatio-temporal indexing method for moving objects. In our former research, we proposed XAT(eXtended Adaptive Tree) structure, consisting of spatial trees and temporal trees, for fast search for spatio-temporal data. The search process in XAT structure is divied into steps. The first step roughly narrows down the potential solutions (moving objects) according to the given search range. The last step fixes the real solution by checking the object's moving track. In this paper, we evaluated the performance between XAT structure and 3D structure for various kinds of search request. Concretely, the computer simulations include: (1) spatio-temporal range search for moving objects. (2) spatio-temporal range search for moving objects' track. (3) nearest-neighbor search for moving objects' track according a given temporal query. We also condeucted to an experiment (1) to compare XAT struture and a new type of 3D structure, which is improved with the above devices done in XAT strucure.

Key words　Spatio-temporal DB, Range search, Nearest-neighbor search, Moving objects

1. はじめに

GPS などの位置を特定可能な機能を備えた携帯端末の普及に伴い、移動オブジェクトの移動データを入手しやすくなりつつある。これらのような時間情報、空間位置情報を持つデータ（以下には時空間データ）を解析し、その結果を将来への対策に役立てるようなニーズが出てくると考えられる。そのような解析を支援する機能の一つとして、ある時間帯にある場所に存在したオブジェクトを検出する機能が考えられる。

筆者らは移動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 XAT 構造（eXtended Adaptive Tree 構造）を提案してきた [2]。XAT 構造は、以前に筆者らが提案した静止オブジェクトに対する時空間データを管理する手法である AT 構造 [1] を、移動オブジェクトを検索できるように拡張したものである。これらの方式では、時空間データの空間情報から作成した木構造（空間木）と、時間情報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ものである。

本稿では、XAT 構造の評価を代表的な時空間データ管理手法である 3D 管理構造 [8] と比較することで行う。また、移動オブジェクトの範囲検索以外の種々の検索要求（移動線分の範囲検索と移動線分の最近接検索）に対しての評価を行うことで、XAT 構造の特性をより深く理解する。

本稿の構成は，2. 節で，筆者らが提案した XAT 構造の概要について説明する。3. 節では，XAT 構造を計算機実験により、移動オブジェクトの範囲検索の観点から、３ D 管理構造と比較し評価する。4. 節では、XAT を種々の検索要求の観点から比較し評価する。最後に，5. 節でまとめを述べる。

2. XAT 構造

2.1 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], [4], [5], [6]。本研究においても，オブジェクトの移動経路データを，この移動線分で表すものとする。

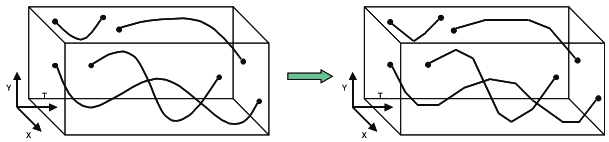


図 1　動物体の移動軌跡の線分化
Fig. 1　Line segments of moving objects.

2.2 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 [1]。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以下の工夫を行う。

（1） 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する。

（2） 上記検査時には，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（実際には二分探索木）で管理する。

移動オブジェクトの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，検索時間が増大する問題がある。これらの工夫はこの問題に対処するためのものである。

また，AT 構造の時間木では，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，二次元平面上の点データに射影し，その点データを管理している。このことも，移動オブジェクトの検索の高速化に貢献すると考えられる。木構造で管理する対象データが一次元直線上の線分データであれば，各中間ノードの分岐条件は範囲となる。すると，異なる中間ノードであっても，その分岐条件の範囲には重なりができることがある。これが二次元平面上の点データであれば，各中間ノードの分岐条件は 2 つのどちらかの軸上での閾値となり，分岐条件に重なりのある中間ノードはできない。この違いは，探索する中間ノード・葉ノード数に影響するため，検索時間にも影響する。AT 構造の時間木におけるこの工夫は，静止オブジェクトだけでなく，移動オブジェクトにおいても有効と考えられる。

2.3 XAT 構造の処理の流れ

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。

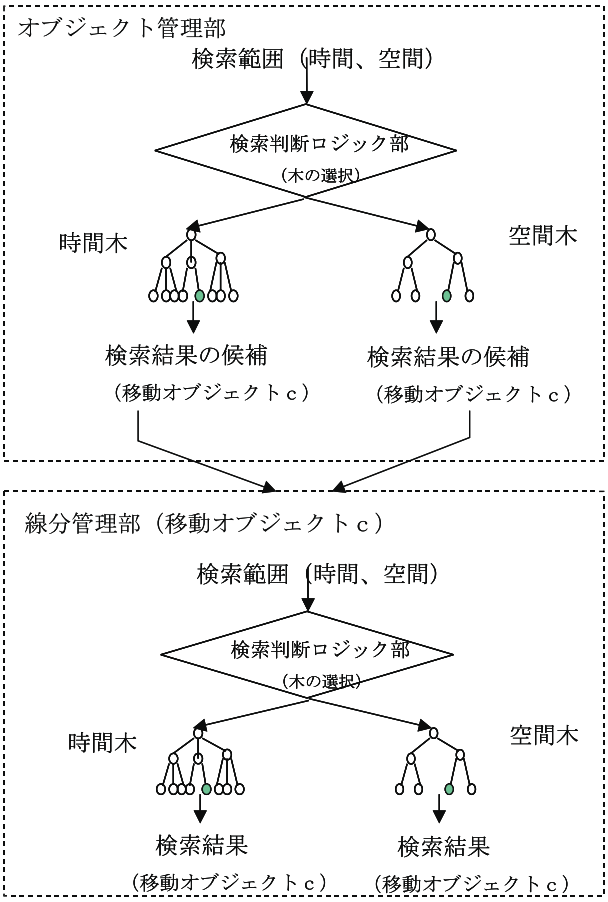


図 2　XAT 構造による検索の流れ
Fig. 2　Flow of search in XAT.

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。

空間木は，オブジェクト（または移動線分）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 (または移動線分) を管理する木構造である。空間木では、各データの中心点を k-d 木 [7] を用いて管理する。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データの外接長方形 (MBR:minimum bounding rectangle) を設定する。

時間木は，オブジェクト（または移動線分）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（または移動線分）を管理する木構造である。ここでは，各データの開始 (発生)・終了 (消滅) 時間を X・Y 軸にとり，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 3 に示すように領域 $y \leqq x$（$x \geqq 0$，$y \geqq 0$）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 3)。

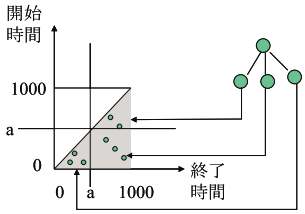


図 3　時間木の構築方法

Fig. 3　Construction of temporal tree.

3.　XAT 構造の評価

3.1　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 構造における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。2. 節でも述べたように，高速化の工夫点は以下の 3 点（AT 構造から引き継いだものを 1 点含む）である。

（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。

このうち，工夫 (2) に関しては，線形リストよりも木構造の方が高速な検索となることは明らかなので，実験により評価をするまでもない。そこで，工夫 (1) と工夫 (3) を行わないバージョンの XAT 構造を，それぞれ XAT 構造 (Type 1)，XAT 構造 (Type 2) として実装し（それぞれの手法の詳細は 3.2 節で説明する），これらとオリジナルの XAT 構造を比較する。

3.2　比較対象手法の詳細

比較対象手法の詳細は次のようになる。

(1) 3D 管理構造

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 4)。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。木構造の中間ノードでは，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 構造のように，オブジェクトの存在時間情報を，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，各中間ノードに，存在時間の範囲情報を分岐条件として設定している（図 5)。

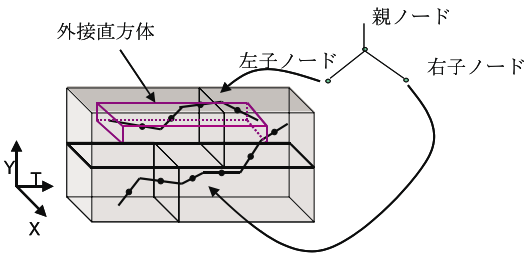


図 4　3D 管理構造

Fig. 4　3D structure.

3.3　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，そのオブジェクト

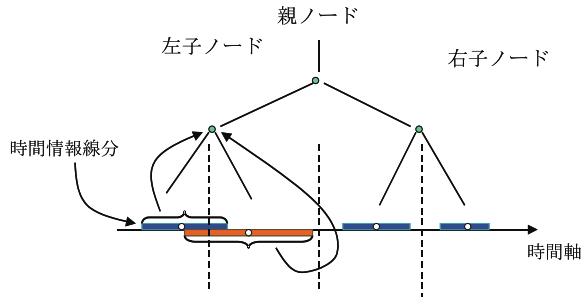


図 5　XAT 構造 (Type 2) におけるオブジェクト用時間木
Fig. 5　Temporal tree for objects in XAT structure (Type 2).

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%

の移動線分群のうち，最も左，右，上，下にあるものを含む長方形の面積としている。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。

3.4　実験方法

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，データ構造を作成し，作成した中間ノード数と作成した葉ノード数を求める。

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) を比較する。ここでは，検索の際に探索した中間ノード数と葉ノード数の和（以下，探索ノード数）を計算する。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) とも比較し，2 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。

3.5　実験結果

3.5.1　データ構造作成

表 2 に，XAT 構造及び 3D 管理構造の作成した中間ノード数と葉ノード数を示す。この表から，XAT 構造より 3D 管理構造の方がこれらの数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。このことから，XAT 構造はメモリコストが悪くなることが分かる。

3.5.2　データ構造検索

(1) **時間検索範囲と空間検索範囲**

図 6-(a),(b),(c) 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探索ノード数を示す。図 6-(d),(e),(f) 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探索ノード数を示す。(a),(d) は直線運動，(b),(e) はランダム運動，(c),(f) は正弦波運動に対応する。なお，図 6 中の 3D 管理構造 (Type 3) については，4.2 節で説明する。

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 管理構造の探索ノード数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 1,2) を比較しても，XAT 構造はいずれの方式よりも，探索ノード数が短いことが分かる。このことから，XAT 構造における高速化の工夫が，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 7-(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探索ノード数を示す。図 7-(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探索ノード数を示す。(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 7-(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，探索ノード数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

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
中間ノード数	32767	62260	133	127	31000	31000
葉ノード数	32768	64271	143	128	32000	32000

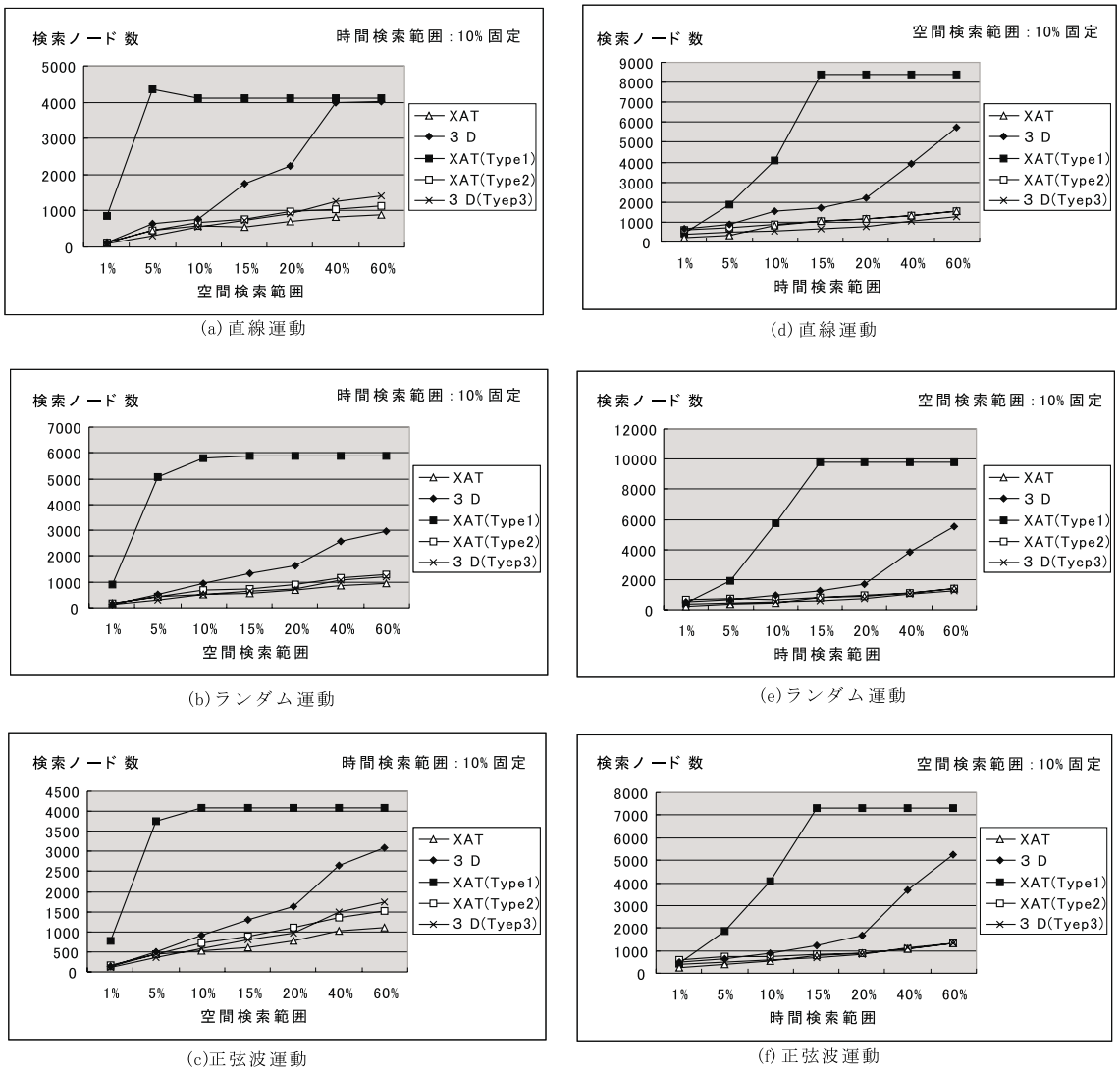


図 6　検索範囲に対する検索コスト

Fig. 6　Search cost for search range.

が増えるため，その探索ノード数が悪化するのである。

図 7-(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 管理構造の探索ノード数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，探索ノード数が悪化する。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，探索ノード数が悪化する。XAT 構造と 3D 管理構造の双方で，探索ノード数が悪化する要因があるため，探索ノード数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。

3.6　結　　論

実験結果から XAT 構造と 3D 管理構造の特性の違いをまとめると，表 3 のようになる。XAT 構造のメモリ消費量は 3D 管理構造の 2 倍になるため，メモリ消費にシビアな応用では，3D 管理構造を選択する必要がある。逆にメモリ環境に余裕のある応用では，オブジェクトの移動範囲が狭い場合には，3D

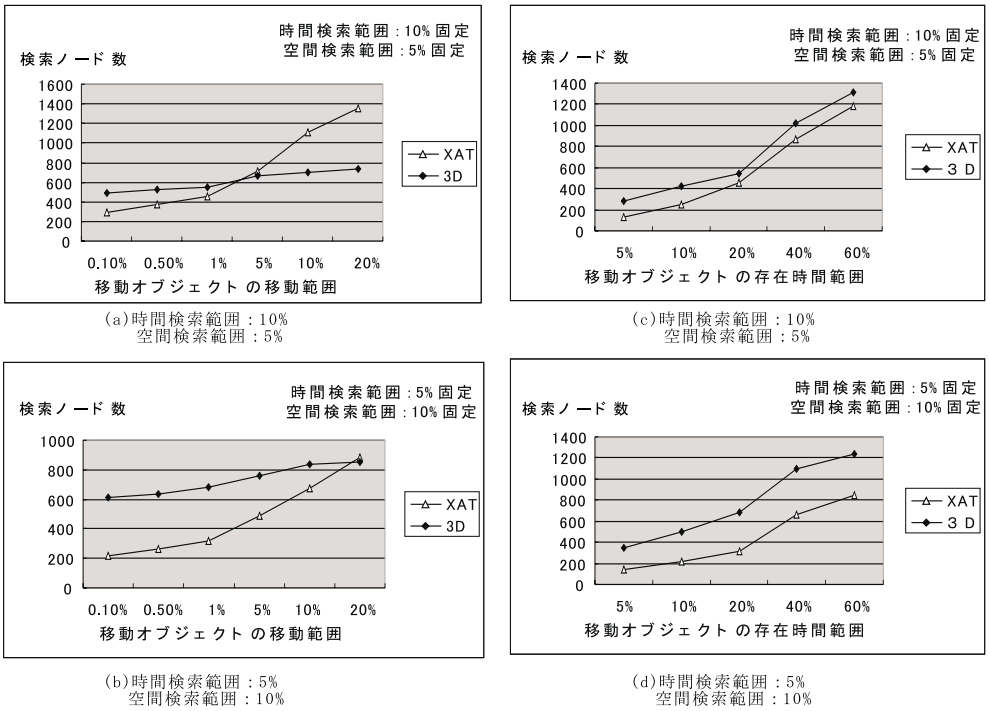


図 7　移動範囲/存在時間範囲に対する検索コスト

Fig.7　Search cost for moving range and existing time.

表 3　XAT 構造と 3D 管理構造の比較

Table 3　Comparison of XAT structure and 3D data structure.

	3D	XAT
メモリ量	○	×
狭い時空間範囲検索	○	○
時間/空間のどちらが広い検索	×	○
オブジェクトの移動範囲が広い	○	×
オブジェクトの移動範囲が狭い	×	○

管理構造よりも XAT 構造の方が優れており，オブジェクトの移動範囲が大きい場合には，XAT 構造よりも 3D 管理構造の方が優れていると言える。また，検索において時間検索範囲と空間検索範囲に差が生じることが頻繁にあることが想定される場合には，3D 管理構造よりも XAT 構造の方が良いと言える。このようにアプリケーションの特性に応じて，XAT 構造と 3D 管理構造を使い分ける必要がある。

XAT 構造の特性に適したアプリケーションとして，ある時点で終了したデータ群に対して分析を行うものが挙げられる。具体的な例としては，消防署や警察署，自衛隊，地域防災センターなどにおける，災害・事件発生時の救急活動状況の事後解析が考えられる。つまり，これらの組織において，救急車両や救急隊員の災害発生時の対応として，いつどこに誰がいたのかを，後々のために災害や事件が起こった後に，人手で解析するような場合である。この応用では，人間が解析するのに，何度も場所や時間を変えて検索しなおすことが想定される。したがって，データがメモリ上に乗る程度のデータ量であっても，何度も検索しなおすことを考えると，一回あたりの検索時間が短い方が良いと言える。その点で，XAT 構造は優れていると考えている。

4.　種々の検索要求と 3D 管理構造の改良

4.1　種々の検索要求に対しての性能比較

4.1.1　目　　的

XAT 構造は，移動オブジェクトの範囲検索を目的として設計されているが，もう少し広い視点から見ると，以下の検索要求も時空間データ管理においては重要となってくる。

（1）　移動線分の範囲検索

（2）　移動線分の最近接検索

ここで，XAT 構造と 3D 管理構造の両検索要求に対する性質を見ることにする。ただし，最近接検索とは，時間検索範囲を固定した空間の最近接検索を指すこととする。そこで，XAT 構造と 3D 管理構造の両方を改良し，移動線分を範囲検索するもの（XAT 構造 (Type 3) と 3D 管理構造 (Type 1)）と移動線分を最近接検索するもの（XAT 構造 (Type 4) と 3D 管理構造 (Type 2)）を実装し，それぞれをシミュレーション実験により比較を行う。

4.1.2　比較対象手法の詳細

比較対象手法の詳細は次のようになる。

(1) XAT 構造 (Type 3)

オリジナルの XAT 構造では，2 段階目の移動線分の木構造を検索する時には，一つでも解が見つかればその時点で検索を終了し，解となる移動線分を持つオブジェクトを検索結果としていた。これに対し，XAT 構造 (Type 3) では，解を見つけても引き続き探索を行い，解となった移動線分を検索結果とする。

(2) 3D 管理構造 (Type 1)

オリジナルの 3D 管理構造では，移動線分検出後に，その移

動線分を持つオブジェクトを検索結果としていたが，3D 管理構造 (Type 1) では，移動線分をそのまま検索結果とする。

(3) XAT 構造 (Type 4)

最初に範囲検索同様，木構造を根ノードから葉ノードへ探索する。葉ノードにたどり着いた時点で，検索キーの空間の一点が含まれる空間範囲は分かるが，最も近い移動線分を得るには，その空間範囲に隣接している範囲の移動線分もチェックする必要がある。そこで，到達した葉ノードから木構造を逆探索し，そこからさらに他のノードを探索する。このときの空間検索範囲は，その時点の最近接の移動線分の距離 d から一辺 $2d$ の長さの正方形を設定する [9]。

(4) 3D 管理構造 (Type 2)

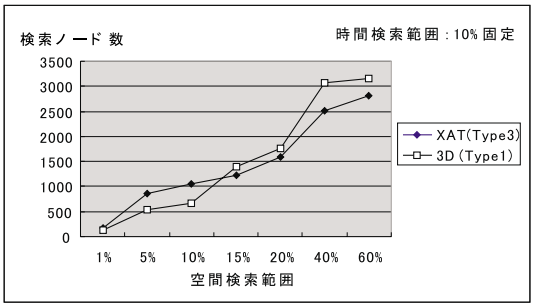
XAT 構造 (Type 4) と同様に，最近接検索を行う。

4.1.3 結　　果

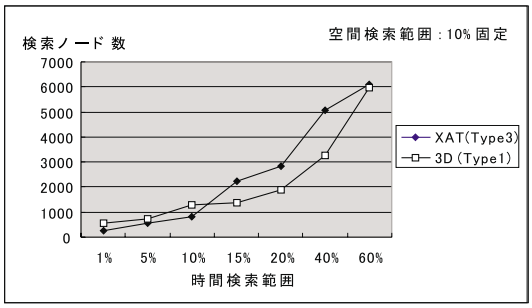
検索要求 (1) に対しての実験は，標準実験条件において時間検索範囲と空間検索範囲を変化させることで行った。図 8-(a) に，時間検索範囲を 10%に固定し，空間検索範囲を 1%-5%-10%-15%-20%-40%-60%と変化させたときの探索ノード数を示す。図 8-(b) に，空間検索範囲を 10%に固定し，時間検索範囲を 1%-5%-10%-15%-20%-40%-60%と変化させたときの探索ノード数を示す。ここでは，直線運動の結果のみ示しているが，ランダム運動や正弦波運動についても，ほぼ同様の結果を得ている。これらの図から，XAT 構造 (Type 3) と 3D 管理構造 (Type 1) では，ほぼ同じ探索ノード数を得ていることが分かる。これに対し，3.5.1 節で説明したオブジェクトの検索の時には，時間検索範囲と空間検索範囲のどちらかの範囲が大きい時に，XAT 構造は 3D 管理構造よりも性能が良かった。オブジェクトを検索の対象にしたときに比べて，移動線分を検索の対象としたときに探索ノード数の差が縮まったのは，XAT 構造のオブジェクトの絞り込みの効果がなくなり，検索範囲中のすべての移動線分を検索したからである。

検索要求 (2) に対しての実験は，標準実験条件において時間検索範囲を，1%, 5%, 10%, 20%, 40%, 60%と変動させて行った。ランダムで開始時刻と空間の中心点の組み合わせを 20 点決め，それらを検索条件として検索を行い，その平均探索ノード数を求めた。その結果を図 9 に示す。まず，3D 管理構造 (Type 2) に着目すると，時間検索範囲が大きくなるほど，探索ノード数が多くなっている。3D 管理構造は中間ノードでは，データを分割するのに，空間の X 軸，空間の Y 軸，時間軸 T の 3 つの軸を順に繰り返して用いている。そのため，時間検索範囲が大きい場合，最初の木構造の探索の時に多くのノードを探索してしまい，探索ノード数が大きくなっている。

次に，XAT 構造 (Type 4) に着目すると，空間検索範囲が大きくなるほど，探索ノード数が少なくなっている。探索ノード数が増えない理由は，空間の最近接検索のために空間木を使っているためで，根ノードから探索していく時に，時間検索範囲は考慮していないためである。また，探索ノード数が減る理由は，時間検索範囲が大きくなるほど，葉ノードから木構造を逆探索した後に，そこからさらに探索する回数が少なくなるからである。詳しく説明すると，XAT 構造 (Type 4) は，葉ノードにたどり着いてから，そこにある移動線分の存在時刻を調べて，それが時間検索範囲に入っているか否かを確かめている。そして，時間検索範囲に入る移動線分のうち，最も中心点の近くにある移動線分までの距離 d を用いて，逆探索してからの探索に用いる空間検索範囲を設定している。時間検索範囲が小さい場合，ある移動線分が中心点から最も近くにあったとしても，その移動線分の存在時間範囲が時間検索範囲に入っておらず，その次に近くにある移動線分をチェックする確率が高くなる。こうして空間検索範囲が大きくなり，逆探索した後に探索するノード数が多くなるのである。これらのことから，時間検索範囲が小さい場合は，3D 管理構造 (Type 2) の方が検索時間は短いが，時間検索範囲が大きくなるほど，XAT 構造 (Type 4) の検索時間の方が短くなることが分かる。



(a)時間検索範囲：10%



(b)空間検索範囲：10%

図 8　移動線分の範囲検索

Fig. 8　Range search of moving segments.

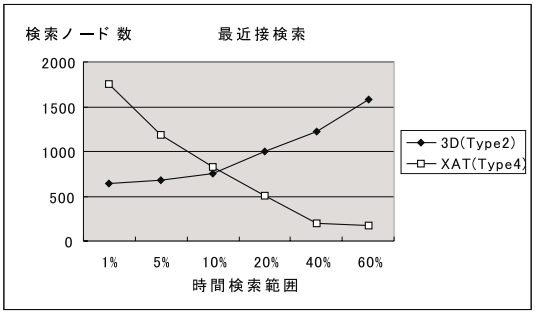


図 9　移動線分の最近接検索

Fig. 9　Nearest neighbor search of moving segments.

4.2 3D 管理構造の改良

XAT 構造は，AT 構造に対して木構造を二重化することで，移動オブジェクトの検索の高速化を狙ったものであるが，同様の工夫は 3D 管理構造に対しても可能である。ここでは，3D 管

理構造を二重化（オブジェクトを管理する 3D 管理構造と移動線分を管理する 3D 管理構造）して改良したもの（3D 管理構造 (Type 3)）を実装し，これと XAT 構造，オリジナルの 3D 管理構造と比較した。実験は標準実験条件において，時間検索範囲と空間検索範囲を変化させることで行った。その結果を図 6 に示す。3D 管理構造も木構造を二重化することで高速化できることが分かる。3D 管理構造 (Type 3) は，空間検索範囲よりも時間検索範囲の方が大きい場合（図 6-(d),(e),(f)）には，XAT 構造とほぼ同じ探索ノード数となっている。しかし，時間検索範囲よりも空間検索範囲の方が大きい場合（図 6-(a),(b),(c)）には，XAT 構造の方が探索ノード数が少なくなっている。このことから，3D 管理構造も木構造を二重化することで，高速化することは可能であるが，それでもなお XAT 構造の方が空間検索範囲が広い場合には，検索時間が早いことが分かった。

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である XAT 構造と代表的な手法である 3D 管理構造をシミュレーション実験により比較評価した。XAT 構造では，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。実験結果から，提案する手法は，メモリコストがかかる欠点はあるが，時間検索範囲と空間検索範囲のどちらか一方が広い場合とオブジェクトの移動範囲が狭い場合は，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。

文　献

[1] 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.

[2] 王軼群, 野澤博, 土方嘉徳, 仲谷美江, 西田正吾, ”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”, 情報処理学会研究報告, DBS-127(FI-67), pp. 73–80, 2002.

[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] 鵜飼規子, 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 dimentional Binary Search Trees Used for Associative Searching,” Comm. of the ACM, Vol. 18, pp. 509–517, 1975.

[8] 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.

[9] 坂内正夫, 大沢裕, ”画像データベース”, 昭晃堂, 1987