ムービングオブジェクトデータベースにおける類似検索機能の改良と評価

北原 由美子† 増永 良文‡

†お茶の水女子大学人間文化研究科数理・情報科学専攻 〒112-8610 東京都文京区大塚 2-1-1
‡お茶の水女子大学理学部情報科学科 〒112-8610 東京都文京区大塚 2-1-1
E-mail: †yumiko@db.is.ocha.ac.jp, ‡masunaga@is.ocha.ac.jp

あらまし 3 次元空間において，物体の動きに関するデータをモーションキャプチャリングシステムである QuickMAG Ⅳを用いて取得し，様々な問合せを実現するムービングオブジェクトデータベースシステムの構築を進めている．これに対する問合せ処理の効率化を目指し，データに索引を付け構造化することにした．本研究では，データベース内のある一つのデータと他のデータ間の相違度を索引とし，B^+木を用いてデータを 1 次元で構造化する索引付け手法に基づいて，実際にこの手法で構造化したデータに対して問合せを行った．データを構造化する以前との性能を比較したところ，この 1 次元索引付け手法が有効であることを確認した．本稿では，数ある問合せ処理の中でも類似検索機能に焦点をあて，この手法とその有効性について論じていく．

キーワード ムービングオブジェクト，時系列データ，類似検索，索引

One-Dimensional Indexing for Similarity Search of Moving Object Database and its Evaluation

Yumiko KITAHARA† Yoshifumi MASUNAGA‡

†Graduate School of Humanities and Sciences, Ochanomizu University
2-1-1 Otsuka, Bunkyou-ku, Tokyo, 112-8610 Japan

‡Department of Information Science, Faculty of Science, Ochanomizu University
2-1-1 Otsuka, Bunkyou-ku, Tokyo, 112-8610 Japan
E-mail: †yumiko@db.is.ocha.ac.jp, ‡masunaga@is.ocha.ac.jp

Abstract In 3-dimensional space, we measure the data about a motion of an object using QuickMAG Ⅳ which is a motion capturing system, and advance construction of a moving object database system which realizes various queries. In order to aim at improvement in the efficiency of this query processing, we organize data with an index. In fact, based on an indexing approach that uses the dissimilarity between one certain data and other data in a moving object database as an index, and structures moving object data with one dimension using B^+-tree, we researched for structured data by this approach. We compared the query performance of no structured data and structured data, and checked that this one-dimensional indexing approach is effective. In this paper, also in some query processing, we focus on a similarity search function and discuss about this indexing approach and its effective.

Keyword Moving object, Time series data, Similarity search, Index

1. はじめに

近年，モーションキャプチャリングシステムや GPS などを用いて，動く物体の位置や姿勢などのデータを計測するセンシング技術が発達している．それに伴い，こうした動きのデータに対して様々な問合せや分析を行いたいという要求が高まっている．そこで我々は，3 次元空間において，物体の動きに関するデータを取得し，様々な問合せを実現するムービングオブジェクト

データベースシステムの構築を進めている．既に，時間データと空間データをデータベース上で統合的に扱うことのできるムービングオブジェクトデータモデルと，それに基づいて，動きに対する問合せを容易にするための問合せ言語である MOQL が提案された[1]．また，動きに関する類似検索機能を実現するために，ムービングオブジェクトデータに対する類似性が体系的に定義され，一部実装された[2]．この類似検索は Query-By-Example の発想に基づいて行われている．さらに，速度・加速度を考慮することにより，同じ軌跡を描く動きであっても，移動速度の変化パターンが異なる動きの違いも認識することのできる類似性も定義された[3]．

本システムの主要な機能である類似検索は，2 つのデータ間の相違度をユークリッド距離関数によって測っている．これまでは類似検索を行うのに，問合せデータを与えたとき，データベースに格納されている全データとの相違度を総当りで計算して，検索結果を返していた．しかし，このままだと格納されているデータ量が多くなるにつれ，検索時間が増加してしまうことが懸念され，データに索引を付け構造化させることが必要となった．これまで数多くの索引技術が研究されているが，代表的なものとして，例えば時系列データをフーリエ変換し，変換後の主要項の係数が成す一般に n 次元の周波数ドメインに写像して，R 木や K-D 木で索引付けする方法がある[6,7]．しかし，変換後の空間は多次元空間となり，現在データベース管理システムのアクセス法として多用されている B 木や B^+木を利用することができない．

そこで，ムービングオブジェクトデータベースにおける類似検索機能の性能の改善を目指し，ムービングオブジェクトデータの索引を 1 次元のスカラー値で与える方法が考案された[4]．これは，データベースに格納されているある一つのデータを基準データとし，その他のデータとの相違度を計算し，得られた値をそのデータの索引として付与することによりデータを構造化する方法である．しかしながら，索引付けすることで実際にはどの程度，類似検索機能の性能の改善がみられるのかの検証はされていない．

そこで本研究では，この索引付け手法についてさらなる議論をし，実際にムービングオブジェクトデータに索引を付け 1 次元で構造化したものに対して類似検索を行い，これまでの総当りでデータ検索を行っていた場合との検索結果を比較し，データを構造化する前後で性能効率はどの程度改善するかについて，検証を試みた．

2. ムービングオブジェクトデータベースと類似検索

先行研究を中心に説明する．

2.1. システムの概要

本研究でのシステム概要を図 1 に表す．ムービングオブジェクトデータの計測には，センシング装置として光学式のモーションキャプチャリングシステムである QuickMAG Ⅳ（応用計測研究所製）を使用する．オブジェクトに 3 点のカラーマーカを付け，ステレオカメラでオブジェクトの動きを 3 次元で計測する．計測後，ユーザは格納インタフェースで計測データのシーンやオブジェクトに関する情報を登録し，挿入エンジンによりムービングオブジェクトデータモデルに従ったデータに変換した上でムービングオブジェクトデータベースに格納される．また，ユーザは問合せインタフェースによって検索エンジンを制御し，Query-By-Example に基づく問合せを行い，検索エンジンを通して検索結果を問合せインタフェース上に返すという流れである．

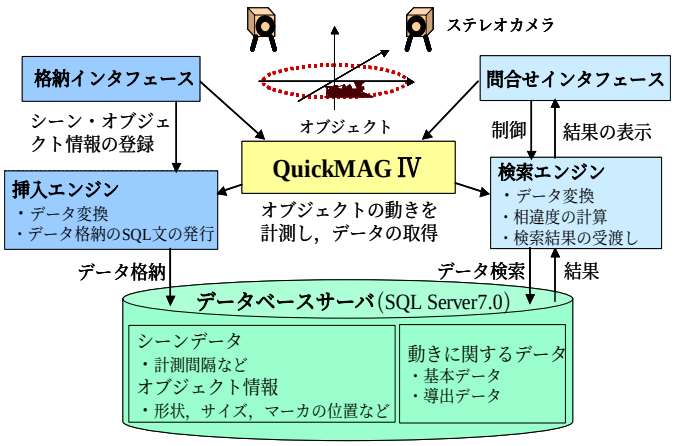

図 1：システム概要

Figure 1：A System Overview

2.2. 動きとは

計測されたオブジェクトの動きは，オブジェクトの中心座標，向き，傾き，およびそれに付与された時刻印の 4 要素で構成される．正式には，計測周波数 f で時刻 t_s から t_e まで計測されたオブジェクトの動きを $\vec{M}=\langle \vec{m}_1,\vec{m}_2,\cdots,\vec{m}_n \rangle$ で表す．このとき $\vec{m}_i=\langle \vec{p}_i,\vec{or}_i,\vec{gr}_i,t_i \rangle$ であり，各要素はオブジェクトの位置ベクトル $\vec{p}_i=\langle x_i,y_i,z_i \rangle$，向きベクトル $\vec{or}_i=\langle ox_i,oy_i,oz_i \rangle$，傾きベクトル $\vec{gr}_i=\langle gx_i,gy_i,gz_i \rangle$ を表している．また，$t_s = t_0$，$t_e = t_n$，$n = (t_e\text{-}t_s) \times f$ である．

2.3. 動きの同一性と基本類似性

動きの同一性の定義について述べた後，動きの各要素の基本類似性の定義について述べる．

【定義 2.1】（動きの同一性）
動き $\vec{M}$ と $\vec{M}'$ が次の条件を満たすとき，2 つの動きは同一であるという．

1. $f = f'$
2. $t_s = t'_s \wedge t_e = t'_e$
3. $(\forall i)\vec{p}_i = \vec{p}'_i$

【定義 2.2】（位置の基本類似性）
動き $\vec{M}$ が $\vec{M}'$ に対して次の条件を満たすとき，位置について ε-類似しているという．

1. $f = f'$
2. $t_s = t'_s \wedge t_e = t'_e$
3. $D_P(\vec{M}, \vec{M}') = \dfrac{\sqrt{\sum_{i=1}^{n} |\vec{p}_i - \vec{p}'_i|^2}}{n} \leq \varepsilon$

【定義 2.3】（向きの基本類似性）
動き $\vec{M}$ が $\vec{M}'$ に対して次の条件を満たすとき，向きについて ε-類似しているという．

1. $f = f'$
2. $t_s = t'_s \wedge t_e = t'_e$
3. $D_o(\vec{M}, \vec{M}') = 1 - \dfrac{\sqrt{1 + \vec{o}r_i \cdot \vec{o}r'_i}}{2n} \leq \varepsilon$

【定義 2.4】（傾きの基本類似性）
動き $\vec{M}$ が $\vec{M}'$ に対して次の条件を満たすとき，傾きについて ε-類似しているという．

1. $f = f'$
2. $t_s = t'_s \wedge t_e = t'_e$
3. $D_g(\vec{M}, \vec{M}') = 1 - \dfrac{\sqrt{1 + \vec{g}r_i \cdot \vec{g}r'_i}}{2n} \leq \varepsilon$

なお，定義から分かるように，これらは開始時刻と終了時刻が同じで，かつ計測周波数が等しい 2 つのムービングオブジェクトデータに対する定義である．そこで，格納データの長さが異なる場合に対応するために，タイムワーピング距離関数を用いてデータ間の相違度を計算する方法や，データ自身を伸縮し長さを等しくした後，ユークリッド距離関数で相違度を計算する方法で類似検索を行った．次に，動きの要素の中でも位置の相違度に着目し，データ長が異なる場合の相違度の計算法について述べる．

2.4. 長さの異なるデータ間の相違度

類似検索を行う際，2 つのデータ間の相違度は距離関数を用いて求められる．主な距離関数としてユークリッド距離とタイムワーピング距離が挙げられる．以下にデータ長が異なる場合の，それぞれの距離関数による相違度の計算法について述べる．

2.4.1. ユークリッド距離

ユークリッド距離関数は等長のデータにのみ適用できるため,データ長の異なるデータを比較する場合,データを等長に揃えてから上記のユークリッド距離関数を用いる．

今，2 つの一般的には不等長な時系列データ $\vec{S} = \langle \vec{s}_1, \vec{s}_2, \cdots, \vec{s}_n \rangle$ と $\vec{T} = \langle \vec{t}_1, \vec{t}_2, \cdots, \vec{t}_m \rangle$ があったとする．$\vec{S}$ を問合せデータとするとき，相違度は次のように求められる．

【定義 2.5】（データの補間）

$$D_P(\vec{S}, \vec{T}) = \frac{\sqrt{\sum_{i=1}^{n} |\vec{s}_i - \vec{t}_i'|^2}}{n}$$

ただし $\vec{t}_i' = \vec{t}_{\lfloor \frac{m}{n} i \rfloor} + \left(\vec{t}_{\lceil \frac{m}{n} i \rceil} - \vec{t}_{\lfloor \frac{m}{n} i \rfloor} \right) \times \left(\frac{m}{n} i - \left\lfloor \frac{m}{n} i \right\rfloor \right)$

ここでは問合せデータに長さを揃え比較対象のデータを伸縮し，比例を用いて比較点を補間している．

2.4.2. タイムワーピング距離

タイムワーピング距離関数[5]はデータ長が異なっていても，データの補間なしで比較することのできる関数である．系列 $\vec{S} = (\vec{s}_1, \vec{s}_2, \cdots, \vec{s}_n)$ を与えたとき，$Head(\vec{S})$ は s_1，$Rst(\vec{S})$ は $(\vec{s}_2, \cdots, \vec{s}_n)$ を表すとする．このとき，非空系列 $\vec{S}$ と $\vec{T} = (\vec{t}_1, \vec{t}_2, \cdots, \vec{t}_m)$ のタイムワーピング距離は次のように定義される．

【定義 2.6】（タイムワーピング距離）

$$D_{warp}(\langle\ \rangle, \langle\ \rangle) = 0$$

$$D_{warp}(\vec{S}, \langle\ \rangle) = D_{warp}(\langle\ \rangle, \vec{T}) = \infty$$

$$D_{warp}(\vec{S}, \vec{T}) = D_{base}(Head(\vec{S}), Head(\vec{T})) + \min \begin{Bmatrix} D_{warp}(\vec{S}, Rst(\vec{T})), \\ D_{warp}(Rst(\vec{S}), \vec{T}), \\ D_{warp}(Rst(\vec{S}), Rst(\vec{T})) \end{Bmatrix}$$

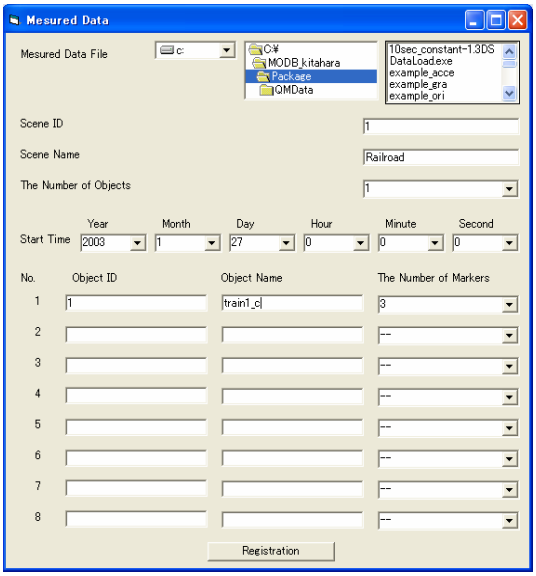

図 2：格納インタフェース
Figure 2：Insert Interface

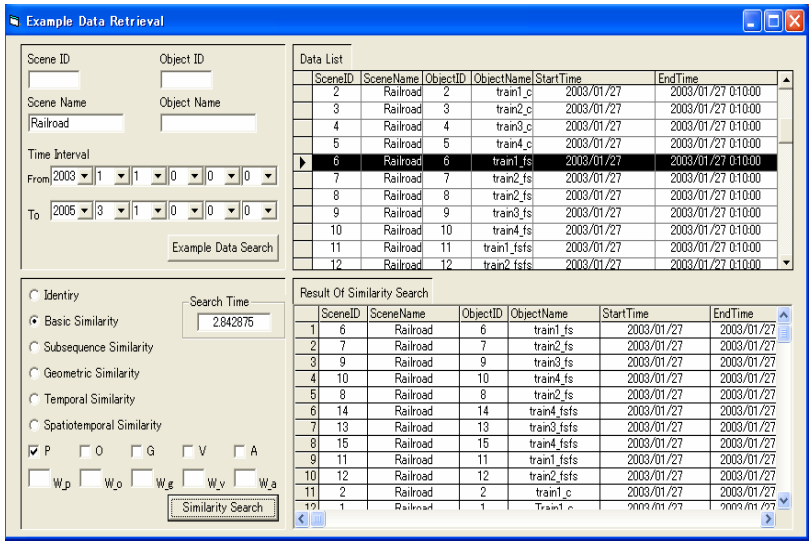

図 3：問合せインタフェース
Figure 3：Retrieval Interface

ここでの＜＞は空系列，D_{base} は 2 つの要素の単純な差の絶対値を表す．

次に，それぞれの距離関数を用いて類似検索を行ったときの検索結果を比較する．

2.4.3. 検索時の比較

ムービングオブジェクトデータには，等速(train_c)，速い－遅い(train_fs)，速い－遅い－速い－遅い(train_fsfs)の 3 つの速度変化パターンで模型列車を走らせた合計 300 のデータを使用する．図 2 に示す格納インタフェースを用いてデータベースに格納したムービングオブジェクトデータに対し，図 3 に示す問合せインタフェースで問合せデータを指定して検索を行う．

表 1 は問合せデータとして train1_c と traim1_fs を与えたときの検索結果をランキングで示したものと，検索に要した時間をそれぞれ示している．

表 1：類似検索結果と検索時間
Table 1：Similarity Search Results and Execution Time

	train1_c			train1_fs	
	ユークリッド	Time Warping		ユークリッド	Time Warping
1	train1_c	train1_c	1	train1_fs	train1_fs
2	train2_c	train2_c	2	train2_fs	train2_c
3	train2_fsfs	train2_fsfs	3	train1_fsfs	train1_c
4	train1_fsfs	train1_fs	4	train2_fsfs	train2_fsfs
5	train2_fs	train2_fs	5	train1_c	train2_fs
6	train1_fs	train1_fsfs	6	train2_c	train1_fsfs
	2.82 (s)	72.95 (s)		2.84 (s)	74.76 (s)

これらから分かるように，タイムワーピング距離を用いた場合，走行パターンがうまく識別されていない．この原因として，ユークリッド距離は同じ時刻の要素同士の差の 2 乗平均であるが，タイムワーピング距離は要素間の差が最小になるものの平均であり，時刻との関連が明確でないという点にあると考えられる．また，タイムワーピング距離を用いると，要素同士の比較回数が膨大になるため，検索に要する時間は平均約 26 倍もかかってしまう．そのため本研究では，長さの異なるデータが存在することを考慮し，ユークリッド距離を用いて類似検索を行うことにする．

しかし，この場合でもデータ数が増加するとそれに伴い検索時間も増加することが懸念される．そこで，上記のように定義されたムービングオブジェクトデータに対して，より効率的に類似検索を行えるようにするために，データに索引を付け，うまく構造化することが重要となってくる．次にデータを構造化する方法について論じていく．

3. ムービングオブジェクトデータの 1 次元索引付け

データを構造化するための技術として，これまで様々な索引手法の研究が行われている．多次元索引構造をもつ代表的な手法では，離散フーリエ変換などの特徴抽出関数を用いて次元を圧縮し，R 木や K-D 木に代表される空間アクセス法を適用する方法が知られている[6,7]．このとき索引には，時系列データを特徴抽出関数により別空間にマッピングした値が使われる．特徴抽出関数による変換は一般に多次元の索引構造を必要としている．

一方，1 次元索引構造である代表的な手法としては，

通常リレーショナルデータベースで用いられている B 木や B^+木があげられる．そこで，既存の DBMS 上で簡単にデータの索引付けおよび類似検索を実行できるようにしたいと考え，ムービングオブジェクトデータを 1 次元空間に変換し，B^+木を用いてデータを 1 次元で索引付けすることにした．次にこの索引付け手法について述べる．

3.1. 相違度に基づく索引付け

まずデータベースに格納されているデータの中からある一つのデータを選定し，それを基準データとする．次に，基準データと他の格納データとの相違度をユークリッド距離関数によって計算する．相違度は 1 次元の値なので，この値を索引として B^+木を用いてデータを構造化する．B^+木はデータの挿入順で木構造が変わってくるが，ここではデータベースに格納されている順にデータを索引付けすることとする．

こうして構造化したデータに対して類似検索を行う，しかし，この基準データとの相違度，つまり各データに付与された索引は相対的なものであるので，実データ同士の距離関係が正確に保存されていない．そのため，本来類似していないデータを検索結果として返してしまう過多誤認が生じる恐れがある．一方，過小誤認は発生しないことが分かっている[4]．そこで次に，索引を利用しながらも過多誤認を排除する類似検索法について説明する．

3.2. 索引を利用した類似検索法

手順Ⅰ．

ある問合せデータ $\vec{Q}$が与えられたとする．まず，基準データ $\vec{R}$と $\vec{Q}$との相違度 $D_p(\vec{R},\vec{Q})$を求める．格納されている全データには，$\vec{R}$との相違度が索引として付与されているのでその値に着目し，

$$D_p(\vec{R},\vec{Q})-\varepsilon \le D_p(\vec{R},\vec{O}') \le D_p(\vec{R},\vec{Q})+\varepsilon$$

となる索引をもつ全てのデータ $\vec{O}'$を検索する．この $\vec{O}'$が全検索結果候補データとなる．

手順Ⅱ．

手順Ⅰで検索された候補データ $\vec{O}'$と問合せデータ $\vec{Q}$との実際のユークリッド距離を計算し，

$$D_p(\vec{Q},\vec{O}) \le \varepsilon$$

となる索引をもつ $\vec{O}$が最終的に $\vec{Q}$と ε-類似している検索結果となる．

例えば，与えられた問合せデータの索引が 4.8 とすると，4.8 に一番近い値の葉ノードにたどり着く．その前後で 4.8±ε に入る探索キーをもつ葉ノードが候補データとなる．また，ε =0 のときは特殊な場合であるが，問合せデータに格納データを用いれば検索は可能である．

3.3. 基準データの選定法

以上に述べた手法により類似検索を行う場合，基準データの選定法が問題となってくる．シミュレーションによりその最適解を探ったところ，データベースに格納されたデータの分布が正規分布に近い偏りである場合，その中心からある程度離れた位置のデータを基準データに取ることで，類似検索時に検証すべきデータ数をある程度抑えられることが明らかになった[4]．そこで，格納されている全データの時刻ごとの座標値を平均して一つのデータを作成し，この平均データ $\vec{A}$との相違度が最も大きいデータを基準データ $\vec{R}$と定めることにする．図 4 は類似検索の検索範囲を，簡単のため 2 次元の点データを用いて表している．問合せデータ $\vec{Q}$と ε-類似しているものを検索するとき，図 4 の環状の網掛け部分に含まれるデータが検索結果候補データ，円状の斜線部分に含まれるデータが最終的な検索結果である．

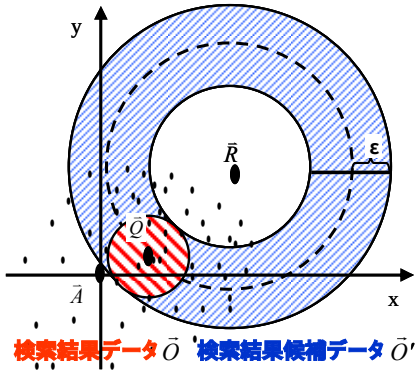

図 4：検索範囲

Figure 4：Search Ranges

データをこの手法を用いて索引付けすることにより，データ間の相違度を計算する回数が減り，検索に要する時間の削減が期待できる．

このように多次元のデータを 1 次元の点に写像し，B^+木を用いて索引付けする手法として，これまでに NB 木[8]や Pyramid Technique[9]という手法が研究されている．しかし，NB 木ではユークリッドノルムという分布にかかわらない固有の値を索引に利用しているため，データの分布によっては効率化が期待できない恐れがある．本研究での 1 次元索引付け手法はデータの分布を考慮に入れ,基準データの選定を行っている．全データの分布が一様分布であった場合，基準データ

の位置に関わらず検索結果の候補データ数は一定となる．したがって，基準データの位置はどこにとってもよいことになり，この場合には本研究と NB 木の手法による検索効率は同じになる．このように本研究の索引付け手法は，NB 木の結果を特殊な場合として含んでいる．

4. 検索時間のモデル化

これまでは問合せデータを与える度にデータ間の相違度を総当りで計算していたが，こうして基準データとの相違度を用いてデータを構造化しておくことにより，類似検索を行った場合，データ間の相違度を計算する回数が減り，検索時間の減少が期待できる．そこで次に，総当りで検索を行った場合とデータを 1 次元索引付けした場合の検索時間を第一次近似でモデル化したものについてそれぞれ論じ,続いて ε =0 とした場合の検索結果について述べる．

4.1. 総当りによる検索時間

ムービングオブジェクトデータが計測順にデータベースに格納されているとする．ある一つの問合せデータを与えたとき，格納データ一つ一つとの相違度を計算するのにかかる平均コストをそれぞれ C_{path0} とすると，全データ数が n のときの検索時間 C_{Ex} は(1)式の通りである．

$$C_{Ex} = C_{path0} \times n + C_0 \quad \text{(s)} \qquad (1)$$

ここで C_0 はデータ数とは無関係な定数項である．

4.2. 1次元索引付けによる検索時間

ムービングオブジェクトデータが前章で述べた方法で B^+木を用いて 1 次元で索引付けされているとする．ある一つの問合せデータを与えたとき，そのデータの索引と一番近い値の葉ノードまでたどるのにかかるコストを C_{path1}，その探索キーから $\pm\varepsilon$ 以内の探索キーをもつ検索結果の候補データを見つけるコストを C_{path2}，過多誤認の検証にかかるコスト，つまり問合せデータと候補データとの相違度を計算するのにかかる平均コストをそれぞれ C_{path3} とする．過多誤認検証の対象となる候補データ数を m_ε とすると検索時間 C_{B+tree} は(2)式のようになる．

$$C_{B+tree} = C_{path1} + C_{path2} + C_{path3} \times m_\varepsilon + C_0' \quad \text{(s)} \qquad (2)$$

ここで C'_0 はデータ数とは無関係な定数項である．

4.3. 同一データ検索 （ε = 0 の場合）

ε = 0 とは同一データ検索を意味し，データベース内のデータを問合せデータとして用いた場合に限り検索可能である．そこで，問合せデータに格納データを用いて同一データ検索を行う場合について考える．

総当りの場合，検索時間 C_{Ex} は ε の値とは無関係なので(1)式と同じままである．一方，1 次元索引構造の場合は，ここではデータベース内のデータは全て異なるものとしているので，m_0 = 1 となり， 検索時間 C_{B+tree} は(3)式のようになる．

$$C_{B+tree} = C_{path1} + C_{path2} + C_{path3} + C_0' \quad \text{(s)} \qquad (3)$$

実際に，検索対象のムービングオブジェクトデータには卓球のスウィングデータを用い検索を行った．データ数 n は 300 個，600 個の 2 通りとし，それぞれに対して類似検索を行う．この中から数個の問合せデータをランダムに抽出し検索を行ったところ，平均検索時間は表 2 のようになった．性能比は 1 次元索引付けすることにより従来の総当り法と比べ，どの程度検索時間が速くなったかを示しているが，索引付けすることで検索時間が著しく改善していることが分かる．

また，データ数が 300，600 のときの総当りによる検索時間をそれぞれ C_{Ex}^{300}，C_{Ex}^{600} とすると，ε = 0 の場合 C_{Ex}^{300} = 6.856，C_{Ex}^{600} = 13.281 となり， (1)式から定数項 C_0 = 0.431，C_{path0} = 0.0214 と決まる．よってデータ数 n が変化してもおおよその検索時間 C_{Ex} を求めることができる．

表 2：同一データ検索の検索時間
Table 2：Execution Time for Identity Data Search

データ数	300data	600data
総当り(秒)	6.856	13.281
1次元索引構造(秒)	0.171	0.338
性能比(倍)	40.1	39.3

今回の同一データ検索では m_0 = 1 であったため, (3)式から分かるように,1 次元索引構造の検索時間 C_{B+tree} は m_ε の値の影響を受けていない. しかし類似検索を行う場合，m_ε の値で検索時間が変化すると考えられる．そこで，実際にデータ数を固定し索引付けしたムービングオブジェクトデータに対し，ε の値を変化させて類似検索を行い，どの程度性能改善がみられるか試みた．次にこの点について説明する．

5. 類似検索の性能評価

卓球のラケットの 3 点にカラーマーカを付け，数名の人にラケットを数回ずつ振ってもらい，そのフルスイングの状態を QuickMAG Ⅳで計測し 600 個のムービングオブジェクトデータを取得する．この計測データを上記の 1 次元の索引構造を用いて構造化したもの

表 3：検索時間（ε = 500）

Table 3：Execution Time（ε = 500）

問合せデータ	data1	data2	data3	data4	data5	data6
総当り(秒)	13.473	13.088	13.204	13.492	13.268	13.183
1次元索引構造(秒)	14.721	14.65	14.416	15.189	15.03	14.862
検索結果数	600	600	600	600	600	600
性能比(倍)	0.915	0.893	0.916	0.888	0.883	0.887

表 4：検索時間（ε = 100）

Table 4：Execution Time（ε = 100）

問合せデータ	data1	data2	data3	data4	data5	data6
総当り(秒)	13.272	13.39	13.663	13.28	13.387	13.274
1次元索引構造(秒)	2.592	3.86	7.602	7.157	4.912	7.99
検索結果数	8	62	256	178	122	210
性能比(倍)	5.12	3.47	1.8	1.86	2.73	1.66

表 5：検索時間（ε = 50）

Table 5：Execution Time（ε = 50）

問合せデータ	data1	data2	data3	data4	data5	data6
総当り(秒)	13.264	13.53	13.274	13.249	13.11	13.205
1次元索引構造(秒)	0.739	2.037	4.793	2.74	1.936	2.861
検索結果数	4	20	114	34	32	44
性能比(倍)	17.9	6.64	2.77	4.84	6.77	4.62

表 6：検索時間（ε = 0）

Table 6：Execution Time（ε = 0）

問合せデータ	data1	data2	data3	data4	data5	data6
総当り(秒)	13.21	13.319	13.503	13.09	13.247	13.469
1次元索引構造(秒)	0.328	0.329	0.343	0.338	0.344	0.341
検索結果数	1	1	1	1	1	1
性能比(倍)	40.3	40.5	39.4	38.7	38.5	39.5

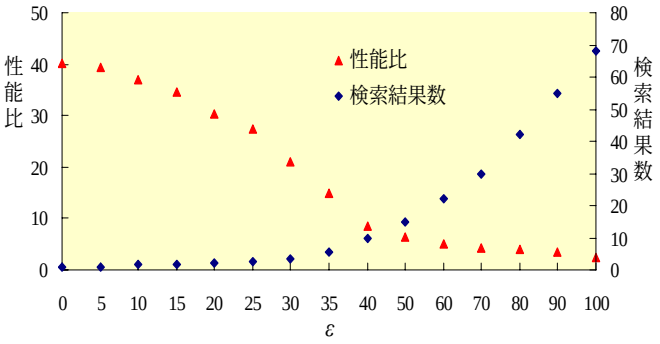

図 5：ε と性能比および検索結果数の関係（データ数 600）

Figure 5：Relation between ε and its performance (600data)

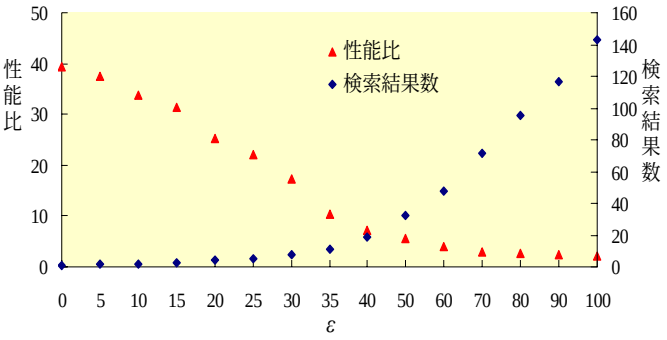

図 6：ε と性能比および検索結果数の関係（データ数 300）

Figure 6：Relation between ε and its performance (300data)

に対して類似検索を行い，検索に要した時間を計測し，従来の総当り法の検索時間と比較する．また，今回は木が最も深くなる最悪の場合を想定し，木のファンアウト数は 2 としている．さらに ε = 0 の検索結果を得るため，問合せデータには格納データを用い，データベースからランダムに抽出し，ε の値を変えて類似検索を行う．

データ数を 600 とし，ε = 500，100，50，0 としたときの検索結果を表 3, 4, 5, 6 に示す．これらの表から，従来の総当りの場合，どんな問合せデータを与えても，また ε がどんな値であっても検索時間はほぼ一定であることが分かる．

一方，1 次元索引付けした場合，問合せデータによって検索時間は異なる．この差は(2)式より候補データ数 m_ε の影響といえる．表 3 より ε = 500 では，索引付けしたほうが平均約 0.897 倍性能が低下してしまっている．この原因としては，検索結果数が全データ数となっていることから，候補データ数が全データ数（m_{500} = 600）となってしまったためと考えられる．

また，検索結果数は ε とともに増減するが，問合せデータによって検索結果数の絞込みに差が生じる．そこで両極端のデータ，ここでは data1 と data3 を除いたデータの平均値に基づいて ε と性能比，および検索結果数の関係を調べたところ図 5 のようになった．同様にデータ数を 300 とし類似検索を行ったところ，検索結果から図 6 の関係が導かれた．

図 5，図 6 より，全データ数の約 5%分のデータを検索結果として取得したいときは，ε = 50 とすればよいことが分かる．このように，ε の値が増加するにつれ性能比は減少していくが，データ数に関わらずこの 1 次元索引付け手法がムービングオブジェクトデータベースにおける類似検索において有効であることが実証された．

また，問合せデータとして yumiko1 を与えたときの検索結果は，総当り法と 1 次元索引付けした場合の両方において表 9 のようになった．検索結果は類似順に並んでいる．これより過多誤認や過少誤認が発生していないことが分かる．

表 9：総当り法および 1 次元索引付けの検索結果

Table 9： Search Result Ranking

Rank	SceneID	SceneName	ObjectID	ObjectName	Dissimilarity
1	251	yumiko001	251	pingpong	0
2	262	yumiko012	262	pingpong	19.2246
3	222	yukari022	222	pingpong	20.1776
4	95	yukari015	95	pingpong	21.2704
5	260	yumiko010	260	pingpong	24.9248
6	94	yukari014	94	pingpong	30.2494
7	53	nakamura003	53	pingpong	31.6042
8	259	yumiko009	259	pingpong	31.6314
9	175	nahoko015	175	pingpong	35.5882
10	264	yumiko014	264	pingpong	37.5081
11	176	nahoko016	176	pingpong	38.0964
12	263	yumiko013	263	pingpong	39.9760
13	173	nahoko013	173	pingpong	43.7946
14	192	ray022	192	pingpong	46.9150
15	96	yukari016	96	pingpong	46.9440
16	180	nahoko020	180	pingpong	47.6504
17	129	yokokawa009	129	pingpong	50.2071
18	196	ray026	196	pingpong	50.6375
19	131	yokokawa011	131	pingpong	53.4804
20	62	ozaki002	62	pingpong	54.1152
21	182	ray012	182	pingpong	55.9660
22	188	ray018	188	pingpong	56.7749
23	54	nakamura004	54	pingpong	57.2358
24	97	yukari017	97	pingpong	57.2407
25	65	ozaki005	65	pingpong	57.6135

6. まとめと今後の課題

本稿では，現行のムービングオブジェクトデータベースにおいてより効率的な類似検索を実現するため，ムービングオブジェクトデータに索引を付け，B^+木を用いて 1 次元で構造化する方法について述べた. また，実際に索引付けしたデータに対し類似検索を行い，従来の総当り法との検索結果を比較し，その有効性について論じた.

今後の課題としては，NB 木や Pyramid Technique のような 1 次元索引構造や，R 木ファミリに代表される多次元索引構造を用いてデータを構造化した場合の検索結果についても合せて比較し，さらに検証する必要がある．また今回の実装では動きの要素である位置，向き，傾きのデータを各々索引付けしたため，各要素に重み付けした検索には適していない．そこで，オブジェクトが回転するなど，全ての要素データが重要となるような複雑な動きのデータに対しても効率的な類似検索が可能となるよう，動きの要素を統合させた索引付けを検討する必要があると考える.

文　献

[1] Y . Masunaga, N . Ukai, “ Toward a 3D Moving Object Data Model -A Preliminary Consideration-, ” In Proceedings of the 1999 international Symposium on Database Applications in Non-Traditional Environments, pp.306-316, November 1999.

[2] 水崎聡子，増永良文，“ムービングオブジェクトデータベースシステムのための類似検索機能の実現に向けて，”情報処理学会データベースシステム研究会報告，vol.2001，no.70，pp.217-223, 2001.

[3] 河内聡恵，増永良文，“ ムービングオブジェクトの速度変化パターンを識別できる類似検索機能の導入，”日本データベース学会 Letters，vol.2, no.1, pp.15-18, May 2003.

[4] 澤井美弥，“ 基準データとの相違度に着目したムービングオブジェクトデータの一類似検索法，” 電子情報通信学会技術研究報告（DBWS2003），vol.103，no.191，pp.247-252, July 2003.

[5] B.-K.Yi, H.Jagadish, and C.faloutsos, “ Efficient Retrieval of Similar Time Sequences Under Time Warping, ” Proceedings of the International Conference on Data Engineering, pp.201-208, IEEE CS, Orlando, FL, February 1998.

[6] Rakesh Agrawal, Christos Faloutsos, and Arun Swami, “ Efficient similarity search in sequence databases, ” In Proceedings of the 4th International Conference on Foundations of Data Organizations and Algorithms (FODO ’93), pp.69-84, Chicago, October 1993.

[7] Davood Rafiei, and Alberto O.Mendelzon, “Querying Time Series Data Based on Similarity,”In Proceedings of the 8th DASFAA, pp.267-274, Kyoto, March 2003.

[8] M.J.Foneseca, and J.A.Jorge, “ Indexing High -Dimensional Data for Dontent-Based Retrieval in Large Databases, ”In Proceedings of the 8th DASFAA, pp.267-274, Kyoto, March 2003.

[9] Stefan Berchtold, Christian Bohm, and Hans-Peter Kriegel, “ The Pyramid-Technique: Towards indesing beyond the Curse of Dimensionality, ”In Proceedings of the International Conference on Management of Data, ACM SIGMOD 1998, pp.142-153, Seattle, Washington, 1998.