

分散型移動軌跡データベースのためのインデクス構築手法

勝田 圭介[†] 柳沢 豊[‡] 佐藤 哲司^{†‡}

携帯電話や車などに GPS が取り付けられ、物体の移動軌跡データの取得が容易になった。これを集めてデータベース化することで、移動する物体の位置を検索できるようになった。近年、センサネットワークの大規模化によってサーバで扱うデータ量が増大し、サーバの負荷が高まっている。そのため、サーバは各センサのもつデータに対するインデクスのみを作成し、データの検索は各センサのもつローカルなデータベースで行う、分散型のデータベースが注目されている。しかし、センサネットワークで使われる伝送路は脆弱であり、各センサがサーバに常時データを送信できないことがある。このため、サーバ上のデータに欠損が生じて、物体の位置をリアルタイムに検索できない可能性がある。これを防ぐためには、データが欠損した場合でも、各センサの持つデータを推定する機能が必要になる。そのため本研究では、サーバが物体の過去の位置や速度から移動先を予測し、センサのもつデータを推定してインデクスを作成する方法を提案した。また、手法の有効性について評価を行った。

Database System for Trajectory of Moving Object on Sensor Networks

Keisuke KATSUDA[§] Yutaka YANAGISAWA[¶] Tetsuji SATOH^{¶§}

In recent years, since there are a number of moving objects which can be tracked by positioning sensors such as GPS, the amount of objects' position data increases the load on the database systems. In order to avoid the concentration of the position data, we have proposed the distributed trajectory database system. The system has the central database server which has only indexes to the position data managed in a local database system for each sensor.

For creating the indexes, the server must receive data from each sensor continuously. However, it is difficult for sensors to send data continuously because the line used in sensor network is so unreliable. Therefore, the database users must wait for the response from the server until the server receive all of necessary data. In order to avoid this problem, we propose several types of predictive indexing methods for moving objects using the past position and velocity of the objects. This indexing method enables the server to process queries using predicted position of moving objects even if the server does not have all of position data. Moreover, we evaluated effectiveness of our proposed indexing methods.

1 はじめに

近年、携帯電話や車のカーナビなどに GPS が取り付けられるようになり、移動軌跡データの取得が容易になりつつある。そして、日常的に誰もが移動軌跡データを扱えるようになった。この移動軌跡データを集めてデータベース化することにより、物体の移動の様子を分析できるようになる。例えば、オフィス内を移動する多くのユーザの移動軌跡を分析すれば、オフィスの構造を最適化するための情報を得られる。同様に、車の移動軌跡データから交通渋滞の予測や、道路整備計画を立てる手がかりが得られる [8]。このほかに、店舗やイベント会場

での来客の行動パターン分析、タクシーやトラックの配車管理 [7]、また動物の生態調査への応用 [5] など、数多くの研究が進められている。

一方、廉価なセンサを大量に環境内に設置しておき、これらを相互にネットワークでつなぐセンサネットワークの研究が進んでいる [2]。ネットワークの各センサが得た全てのデータをサーバに集めることで、従来よりはるかに大量のデータを集めることができ、より詳細にデータを分析できるようになった。しかし、設置されるセンサの数が大量になると、センサネットワークが大規模化することで、一つのサーバで全てのデータを一元的に管理することが難しくなってきた。

この解決法として、データを分散して管理する分散型のセンサデータベースが提案されている [10]。分散型センサデータベースでは、データは分散配置された各センサのローカルなデータベースが保持し、各センサはロー

[†]大阪大学 大学院情報科学研究科情報ネットワーク学専攻

[‡]日本電信電話株式会社 NTT コミュニケーション科学基礎研究所

[§]Graduate School of Information Science and Technology, Osaka University

[¶]NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation.

カルなデータベースからデータの一部をサーバに送信する．サーバは受信したデータを用い、各センサのもつローカルなデータベースに対するインデックスをつくる．サーバはユーザからのクエリを受け取ると、まずインデックスを用いて検索すべきデータを保持するセンサを特定し、次にそれらのセンサのローカルなデータベースに対してクエリを転送する．そして、ユーザは求めるデータを得ることができる．

しかし、一般にセンサネットワークにおいては、データの伝送路に無線ネットワークが用いられるなど、データを送受信する伝送路が脆弱で不安定である場合が多い．このため、各センサは常時サーバへ安定的にデータを送信できない可能性がある．データの欠損や遅着が生じると、サーバでのインデックス作成に遅延が生じ、結果的にユーザからのクエリに対して、リアルタイムに物体の位置を検索できないケースが生じる．これを防ぐためには、物体の将来の位置をある程度予測してインデックスを作成する方法が有効である．この方法を用いることで、未着分のデータに対する検索要求にも高速に応じることができる．センサデータの中でも、移動軌跡データについては物体が移動するときの物理的制約から、物体の将来の位置を過去の移動軌跡データから推測し、比較的単純な時間関数により表すことができるという性質がある [6]．この性質を利用して、サーバが各センサから受信する各物体の時刻、位置、速度を用いて物体の将来の位置を表す時間関数を求め、この関数に基づいた予測型インデックスを作成する方法が利用できる．

以上のことから本研究では、分散型データベースにおいて、物体の位置、速度に着目して作成するいくつかの予測型インデックス手法を提案する．そして、提案したいくつかの予測型インデックス手法を検索効率や予測精度などの観点から比較するための、分散型移動軌跡データベースシステムを実際に作成して評価を行う．評価用のデータとして、実際に GPS を用いて取得した、奈良市内で営業する人力車のべ 39 台分の移動軌跡データを用いる．

以下、まず 2 章で本研究で対象とする分散型移動軌跡データベースについて触れ、3 章で本研究で提案する予測型インデックス手法の内容について説明する．さらに、4 章で実装した実験用システムを用いて、提案したいくつかの予測型インデックス手法の比較実験を行い、最後に 5 章で本研究のまとめを行う．

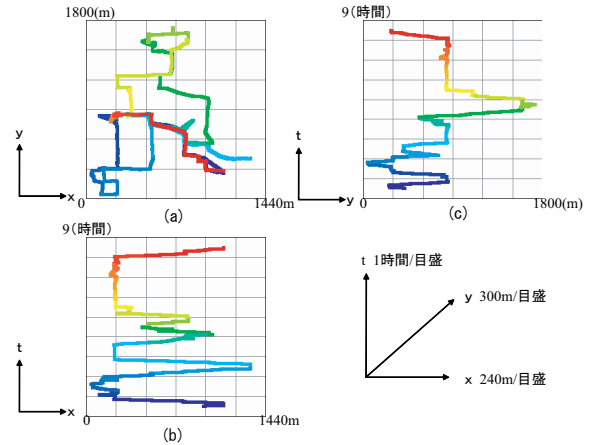


図 1: GPS を用いて取得した移動軌跡データ

2 分散型移動軌跡データベース

2.1 移動軌跡データ

本研究で用いる移動軌跡のためのデータモデル [9] について概略を示す．実世界中の移動する物体の移動軌跡は、時間 t から空間 R^n への連続な関数として表すことができる [3]．しかし、センサなどを用いて物体の移動軌跡を取得すると、実際には離散的な値しか得られない．このため、本研究では、移動軌跡データは不定間隔でサンプリングされた座標値の列としてモデル化し、連続化が必要な場合は下記の近似関数 $\tilde{\lambda}(t)$ を用いる方法をとる．

ある移動するオブジェクトの識別子を o とし、時刻列を $T = \langle t_1, t_2, \dots, t_m \rangle$ と表し、 n 次元空間 R^n 上のベクトルを $p = (x^{(1)}, x^{(2)}, \dots, x^{(n)})$ とする．このとき T の各時刻に記録されたオブジェクト o の n 次元空間上の位置を表すベクトル列を $P = \langle p_1, \dots, p_m \rangle$ とすると、オブジェクト o の移動軌跡データ λ は (o, T, P) の三つ組として表すことができる．なお $\lambda(t_i) = p_i$, $|\lambda| = m$ とする．

また、本研究では簡単化のため移動軌跡データは、特に断らない限り 1 秒間隔で測定されているものとする．つまり $t_{i+1} - t_i = 1\text{sec}$ である．実際に GPS を用いて取得した移動軌跡データの例を図 1 に示す．これは、奈良市内で営業する人力車に GPS 受信機を取り付けて取得した、1 台の 8 時間分の連続した移動軌跡データである．図 1(a) は移動軌跡データを平面上にプロットしたものである．図 1(b)、図 1(c) はそれぞれ時間軸を縦にとり、 x 軸、 y 軸の動きを表したものである．

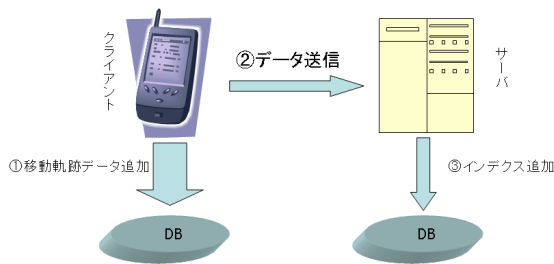


図 2: 分散型移動軌跡データベース (データ追加)

2.2 移動軌跡データベースと分散管理

移動軌跡データベースとは、各センサで取得したオブジェクトの移動軌跡データを全てサーバに送信し、サーバが受信した全てのオブジェクトの移動軌跡データを管理するデータベースのことである。このデータベースにユーザが、時間、位置、物体の ID などの条件を指定したクエリを送ると、サーバがその条件を満たすようなデータを、サーバが保持している全ての移動軌跡データを用いて検索する。このデータベースでは、サーバが全てのオブジェクトの移動軌跡データを管理する必要がある。しかし、近年のセンサネットワークの大規模化によりセンサデータが増大し、サーバで管理するデータ量が莫大になりサーバの負荷が高まってきた。今後、センサネットワークが拡大するにつれて、サーバにより一元的にセンサデータを管理する方法が限界を超えることは確実である。

このサーバの破綻を回避する方法として、データを分散して管理する分散型のセンサデータベースが提案されている [10]。分散型センサデータベースでは、データは分散配置された各センサのローカルなデータベースが保持し、ローカルなデータベースはデータの一部のみをサーバに送信する。サーバは送られてきたデータを用い、各センサのもつローカルなデータに対する大まかなインデクスをつくる (図 2)。サーバはユーザからのクエリを受け取ると、まずインデクスを用いて検索すべきデータを保持するセンサを特定し、次に実際にデータを保有しているセンサのローカルなデータベースに対してクエリを転送する。それぞれのデータベースでは、蓄積しているデータに対してクエリの処理を行い、答えとなるデータを見つけ出す。そして、ユーザは求めるデータを得ることができる (図 3)。

しかし、一方でデータベースを分散化することにより、サーバは各センサのもつデータをリアルタイムに把握することができなくなり、クエリの即応性の低下が生じて

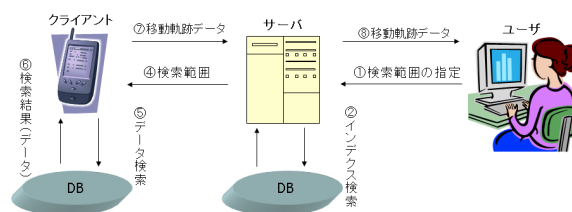


図 3: 分散型移動軌跡データベース (データ検索)

しまう。これは、サーバは各センサからデータを受け取るまでは各センサが現在保有するデータを知ることができないからである。例えば、各センサは 5 分に 1 回しかデータをサーバに送信しないとすると、もしユーザから「現在の範囲 A に存在する物体は?」というクエリがやってきても、サーバがクエリの発生した時刻のデータを、全てのセンサから受信していなければ、サーバはその時刻の物体の位置を把握できない。つまり、最大 5 分間待ってサーバが全てのセンサからその時刻のデータを受信するまで、ユーザのクエリに答えることはできない。

このクエリの即応性を失わないようにするために、本研究では各センサから送られてきた過去のデータから、ある程度未来のデータを予測する方法を導入する。先ほどの例においてこの方法を導入すると、過去のデータを用いてクエリが発生した時刻における全ての物体の位置を予測し、範囲 A に存在すると予測した物体の移動軌跡データを保持するセンサに、クエリを転送する。これによりサーバは、全てのセンサからクエリ発生時刻における物体の移動軌跡データを受信していなくても、ユーザのクエリに答えることができる。

このようにデータを分散化することで、サーバで管理するデータ量を削減でき、各センサは常時サーバにデータを送信しなくて済む。この分散化データベースにおいては、サーバが各センサから受信する移動軌跡データを用いてどのように予測を行うかが一番の鍵となる。次章でその予測手法について述べる。

3 予測型インデクス手法

本章では、本研究で提案する移動軌跡データに対する予測型インデクス手法について述べる。予測型インデクスとは、従来の空間インデクスに時間経過によるオブジェクトの移動先を予測する手法を導入したものである。具体的には、一般の空間インデクスにおける MBR (Minimum Bounding Rectangle) [4] が一定時間経過後にどのような形に変形するかを、MBR に含まれるオブ

ジェットの情報から予測する方法を用いる。

効率よく目的のセンサを検索するためには、MBR の面積に含まれるオブジェクトの個数の存在密度を高くし、なおかつ MBR の総数を減らす必要がある。本研究では、規定の大きさのグリッドをつかってオブジェクトをグルーピングして MBR を作成する方法を用いる。一方、センサから受信するデータ量を減らすためには、長時間先までの MBR の変形を予測する必要がある。本研究では、オブジェクトの過去の瞬間速度、平均速度、向き、最大速度を用いたオブジェクトの移動予測法を導入することで、これを実現する。

以下、これらを実現するためのオブジェクトのグループ化手法、オブジェクトの移動予測法について述べる。

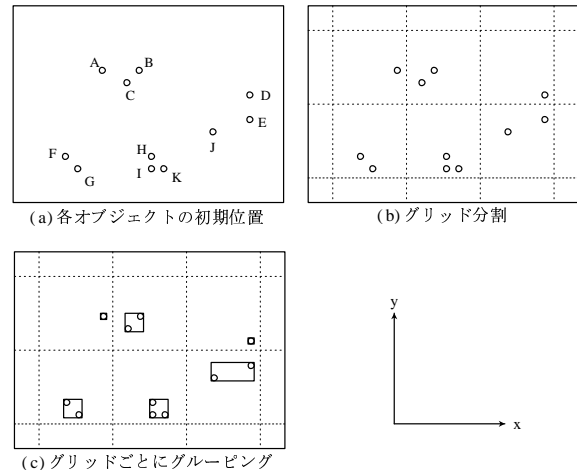


図 4: グルーピング

3.1 オブジェクトのグループ化手法

初期位置が近いオブジェクトは、それ以降も同じように移動することが多い [1]。そこで、サーバが各センサから移動軌跡データを受信するたびに、各オブジェクトの位置を調べて、近くに存在するオブジェクト同士をグルーピングして MBR を作る方法を用いる。これにより、移動の様子が似ているオブジェクトを同じ MBR に含めることができる。以下、このグルーピングについて説明する。

まず、移動軌跡データが存在する範囲をあらかじめグリッドを使って分割しておく。移動軌跡データを受信した時に、同じグリッドに含まれるオブジェクトをグルーピングする。例えば、図 4(a) のように A ~ K のオブジェクトの移動軌跡データを各センサから受信したとする。まず、図 4(b) のように全領域をグリッドに分割し、同じグリッドに含まれるオブジェクトをグルーピングする。これにより 11 個あったオブジェクトは 6 つのグループにまとめられる。次に、図 4(c) のように、同じグループに含まれるオブジェクトを全て含む最小の矩形領域を作る。このように、グリッドを用いてグルーピングすることで初期位置の近いオブジェクトを高速にグルーピングできる [6]。

3.2 オブジェクトの移動予測法

本節では、前節で述べた初期 MBR が j 秒経過したときに、どのような形に変形するかを予測する方法を述べる。以下、サーバがセンサからデータを受信した時点で作成する MBR を「初期 MBR」、予測法により j 秒後

の変形を予測した MBR を「予測 MBR」、 j 秒後のオブジェクトの真の位置を使って作る MBR を「真の MBR」と呼ぶ。

真の MBR は予測 MBR の精度を測定するために用いるだけで、実際にサーバで作ることはできない。なぜなら、サーバは各センサから常時データを受信できないからである。例えば、サーバが各センサから移動軌跡データを 60 秒間隔で受信する場合は、60 秒に 1 回しか各オブジェクトの位置を知ることができない。つまり、一度 MBR を作ったとしても 60 秒後まで MBR を作り直すことができないことになる。仮に MBR を作った 10 秒後にオブジェクトの位置を知りたくても、MBR を作成した後で、オブジェクトは移動した可能性があり、過去に作った MBR はそのままでは検索に利用できない。そこで、過去の各オブジェクトの動きからそれぞれのオブジェクトがどのように移動するかを予測して、 j 秒後の予測 MBR を作る。この予測 MBR が実際のオブジェクトの位置を正確に予測できていれば、これをインデクスとして利用することができる。以下、この予測 MBR を求める方法として、オブジェクトの速度を用いる方法をいくつか提案する。尚、予測精度に関しては 4 章で詳しく述べる。

まず準備として、初期 MBR ごとにそれぞれに含まれるオブジェクトの x 方向、 y 方向、正方向、負方向別にそれぞれ最大速度を求め、 x 軸正方向の最大速度を v_{+x} 、 x 軸負方向を v_{-x} 、 y 軸正方向を v_{+y} 、 y 軸負方向を v_{-y} とする。また、各センサからサーバに移動軌跡データが送られてきた時刻を t_s 、ユーザからクエリが送られてきた時刻を t とする。これらを用いて、初期 MBR から x 軸正方向に $|v_{+x}|(t-t_s)$ 、 x 軸負方向に $|v_{-x}|(t-t_s)$ 、 y

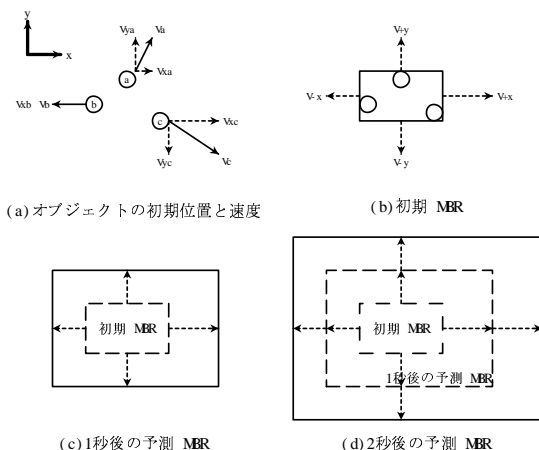


図 5: MBR の変形の予測例

軸正方向に $|v_{+y}|(t - t_s)$, y 軸負方向に $|v_{-y}|(t - t_s)$ だけ変形したものを予測 MBR とする。

例えば, 図 5(a) のように 3 つのオブジェクトの移動軌跡データを受信したとする。図中の, 矢印の出発している位置がそのオブジェクトの位置で, 矢印の大きさが速度を表している。まず, 図 5(b) のように三つのオブジェクトをグリッドを用いてグルーピングし, a, b, c を全て含む最小の矩形領域である初期 MBR を決める。また $v_{+x} = \max(v_{xa}, v_{xc})$, $v_{-x} = v_{xb}$, $v_{+y} = v_{ya}$, $v_{-y} = v_{yc}$ のように四方向別にそれぞれ最大速度を求める。ここで予測 MBR を作るのに必要な初期 MBR と方向別の最大速度が求められたので, 各センサから移動軌跡データを受信してから 1 秒後の予測 MBR は図 5(c), 2 秒後の予測 MBR は図 5(d) のようになる。

予測 MBR を作る際の要因となる速度に関して, 本研究では, 瞬間速度, 平均速度, 最大速度を用いた。以下, それぞれについて説明する。尚, 各センサは毎秒位置情報をセンシングし, j 秒ごとにサーバへ移動軌跡データを送信するものとする。

1. 瞬間速度

各センサが移動軌跡データをサーバへ送信する際の, 各オブジェクトの速度である。各センサは, サーバにデータを送るとき (j 秒に 1 回) のみ瞬間速度を求める。速度情報は正方向負方向に分ける必要がなく, データは 2 方向 (x 軸方向, y 軸方向) ですむ。しかし, 各センサがサーバにデータを送信するその一瞬の速度しか考慮に入れていないため, 極値のような速度をとる瞬間にデータを送信すると, 以後のオブジェクトの移動軌跡と大きくずれが生じ, 結果

として予測 MBR の予測精度が落ちる。

2. 平均速度

求める平均速度を v_{ave} , 各センサが移動軌跡データをサーバに送信してから i 秒後におけるオブジェクトの瞬間速度を v_i とすると,

$$v_{ave} = \sum_{1 \leq i \leq j} \frac{v_{ave} + v_i}{2}$$

により与えられる重み付き平均速度である。平均速度を正方向, 負方向に分けて求めるため, 各センサがサーバへ送る速度情報のデータ量が瞬間速度の時に比べると二倍になる。また, 最近の速度に対し重み加わるため, ある程度の速度の変動には柔軟に予測 MBR が変化し, 実際の移動軌跡にマッチするが, 速度の変動が激しいオブジェクトだと, 予測 MBR の予測精度が低下したり, 逆に冗長すぎる予測 MBR を生じたりする。また, 各センサはサーバに移動軌跡データを送らない間も含め毎秒, 瞬間速度, 平均速度を計算する必要があり, 各センサの負荷が高くなる。

3. 最大速度

求める最大速度を v_{max} , 各センサが移動軌跡データをサーバに送信してから i 秒後のオブジェクトの瞬間速度を v_i とすると,

$$v_{max} = \max_{1 \leq i \leq j} v_i$$

で与えられる瞬間速度の最大の速度である。各センサがサーバに移動軌跡データを送った後に毎回, 最大速度を 0 にリセットする方法と, リセットせずにそのまま最大速度を累積する方法がある。リセットする方法では, オブジェクトが減速を続ける場合に, 予測 MBR の冗長部分を削除できる。しかし, 急な加速に対応できずに予測 MBR でカバーできないオブジェクトが生じる可能性がある。一方, リセットしない方法では, 予測精度を高く保つことができるが, 各オブジェクトが最大速度よりも大きく下回る速度で長時間移動する場合, 予測 MBR に多くの冗長部分が生じる。

また, 速度は四方向 (x 軸正方向, x 軸負方向, y 軸正方向, y 軸負方向) に存在する。このため, 各センサにおいて四方向それぞれ別に最大を求める方法と, 方向の区別なく, 全方向共通の最大を求める方法がある。前者に比べ, 後者は速度が四方向同じ

値となるので、各センサからサーバに速度を送る際のデータ量が $1/4$ となる。また、予測 MBR が初期 MBR から四方向に同じだけの変位となるので、予測 MBR に矩形でなく円 (Minimum Bounding Sphere) を用いることもでき、円を用いることで予測 MBR の冗長部分を削ることができる。

4 性能比較実験

本研究では、3章で提案した予測型インデクス手法を比較、評価するための実験用システムを実装した。本章では、実際に GPS を用いて取得した移動軌跡データを用いて、手法の比較評価実験を行った結果について述べる。

4.1 評価指標

まず、提案した予測型インデクス手法を比較評価するための指標について述べる。本研究で対象とする分散型移動軌跡データベースは、センサネットワーク上におけるアプリケーションを想定している。そのため、省電力の観点からセンサデータの送信回数、通信量はできるだけ抑えたい。つまり、少ない送信データ量で高い予測精度を得たい。一方、高速に効率よくオブジェクトの検索もできる必要がある。このようなことから、以下の3つの評価指標を挙げる事ができる。

1. 送信データ量

各センサからサーバへ送信するデータ量 (byte):
 $d_s a n / i$

2. 予測精度

予測 MBR に含まれる移動軌跡データの割合 (%):
 $(d_{in} / a d_s) * 100$

3. 検索効率

予測 MBR 面積に対する、真 MBR 面積の割合 (%):
 $(S_p / S_i) * 100$

表 1 に説明用の変数を定める。ここで、各センサからサーバへ送信する移動軌跡データに含まれる時刻、位置、速度のデータサイズのそれぞれを 1 (byte) とする。また、予測手法によって一度にサーバに送信する移動軌跡データ量 (n) が異なる。例えば、瞬間速度を用いて予測 MBR の大きさを決める手法のときは $n = 5$ (時刻, x 座標, y 座標, x 方向速度, y 方向速度), 平均速度, 四

表 1: 変数定義

オブジェクト数 (個)	a
サーバへの移動軌跡データの送信間隔 (秒)	i
一度にサーバに送信する移動軌跡データ量 (byte)	n
予測 MBR に含まれる移動軌跡データ量 (byte)	d_{in}
一つのオブジェクトの移動軌跡データ量 (byte)	d_s
予測 MBR 面積 (m^2)	S_i
真の MBR 面積 (m^2)	S_p

方向それぞれで最大を求めたときの最大速度を用いて予測 MBR の大きさを決める手法のときは $n = 7$ (時刻, x 座標, y 座標, x 軸正方向速度, x 軸負方向速度, y 軸正方向速度, y 軸負方向速度), 四方向共通の最大を求めたときの最大速度を用いて予測 MBR の大きさを決める手法のときは $n = 4$ (時刻, x 座標, y 座標, 速度) のようになる。

本実験においては、以上のような評価指標を用いて、送信データ量の大小における、予測精度、検索効率が悪化したインデクス手法を調べた。

4.2 実験方法

まず、本研究で実装した実験用システムについての概要を説明する。移動軌跡データは毎秒、各センサのもつデータベースに追加される。そして、各センサは一斉にある一定間隔でサーバに、時刻、位置、速度を送信する。サーバは受信したデータを基に、どのセンサがどんなデータをもつかをインデクスを用いて管理する。

ユーザが、検索したい移動軌跡データの範囲を指定したクエリをサーバに送ると、サーバはインデクスを用いて、その範囲にある移動軌跡データをもつセンサを探し、該当するセンサ全てにクエリを転送する。クエリを受け取ったセンサはローカルなデータベースを用いてクエリを実行し、結果をサーバに返す。サーバは各センサから受信した結果をまとめてユーザに送ることで、ユーザの要求が満足される。

以上のような実験用システムを用いて、インデクス手法間の性能比較実験を行った。用いたデータは実際に GPS を用いて取得した、奈良市内で営業する人力車のべ 39 台分の移動軌跡データで、その中からシステムがランダムに 10, 20, 30 個 ($a = 10, 20, 30$) 選び、測定は 10 回行い、その平均値を測定値とした。それぞれのオブジェクトで読み込む移動軌跡データの数は 2 万個 ($d_s = 20000$) とする。評価対象となるインデクス手法

表 2: 評価対象手法

	速度	グルーピング	リセット	最大速度	形
a	瞬間	有	-	-	-
b	平均	有	-	-	-
c	最大	有	有	異	-
d	最大	有	有	共通	円
e	最大	有	有	共通	矩形
f	最大	有	無	異	-
g	最大	有	無	共通	円
h	最大	有	無	共通	矩形
a'	瞬間	無	-	-	-
b'	平均	無	-	-	-
c'	最大	無	有	異	-
d'	最大	無	有	共通	円
e'	最大	無	有	共通	矩形
f'	最大	無	無	異	-
g'	最大	無	無	共通	円
h'	最大	無	無	共通	矩形

は、予測 MBR を決めるのに用いる速度の種類、グルーピングの有無、最大速度のリセットの有無、最大速度を四方向共通にするか否か、予測 MBR として円を用いるか、により 16 種類に分けられる。これを表 2 にまとめる。尚、グルーピングを行う際のグリッドの個数は 25(5 × 5) 個とした。

まず、インデクス間の違いを明確にするため、送信データ量を固定 (10000 byte) したときの予測精度、検索効率を求めた (図 6, 7)。次に、送信データ量の変化に対する予測精度、検索効率を評価するため、各センサからサーバへの送信間隔を 2,4,10,20,50,100,200,500,1000 と変化させながら、その時の送信データ量に対する予測精度、検索効率を測定した。このとき、図 6, 7 より、予測精度、検索効率ともにほとんどの手法において、グルーピングを行う方が、グルーピングを行わないよりも優れているという結果が出たことと、また、d, e と、f, g, h は結果が類似していたことより、手法 a, b, c, e, h についてのみ、送信データ量に対する予測精度、検索効率をグラフ化した (図 8, 9)。

4.3 実験結果

図 6, 7, 8, 9 を元に、送信データ量に基づいて、予測精度、検索効率の優れた予測手法について考察する。

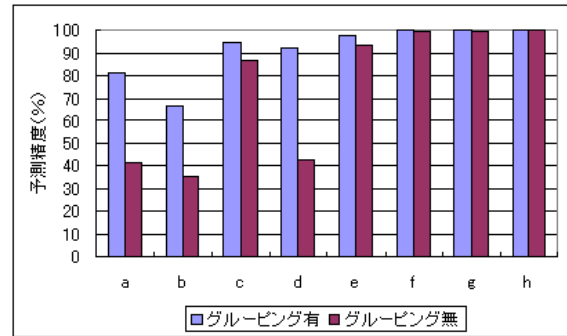


図 6: 手法間の比較 (予測精度)

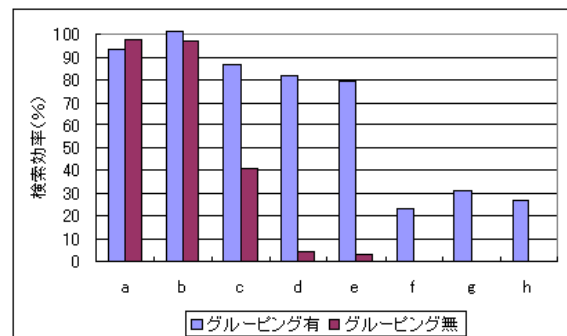


図 7: 手法間の比較 (検索効率)

送信データ量が極端に少ない場合、予測精度、検索効率がどちらもよいというのではない。例えば、a, b は検索効率はよいが、予測精度が低くなる。逆に、c, e, h は予測精度は高いが、検索効率が悪くなる。送信データ量が比較的少ない 10000 byte の場合、c, e が予測精度、検索効率ともに優れたインデクス手法といえる。送信データ量が多くてもしっかりした場合、全ての手法の検索効率がほぼ 100% になるので、予測精度が高い h が優れた予測手法といえる。つまり、送信データ量を大きくすると、予測精度、検索効率ともに大きくなるが、送信データ量をどれぐらい許容するかによって、最適なインデクス手法は異なることが分かった。

グルーピングを行わない場合、1 つのオブジェクトに対し 1 つの MBR を割り当てるため、結果的に多数の MBR が作られ、オブジェクト間で予測 MBR の重なりが生じる。このため、検索効率が著しく低下したと考えられる。これに比べると、初期位置のみを考慮したグルーピングでも、効率のよいインデクスを作成できている。また、最大速度のリセットの有無によっても検索効率は大きく異なる結果となった。最大速度のリセットはデータの性質によって結果が左右される傾向がある。今回の実験のように、予測 MBR の冗長部分を削ることが

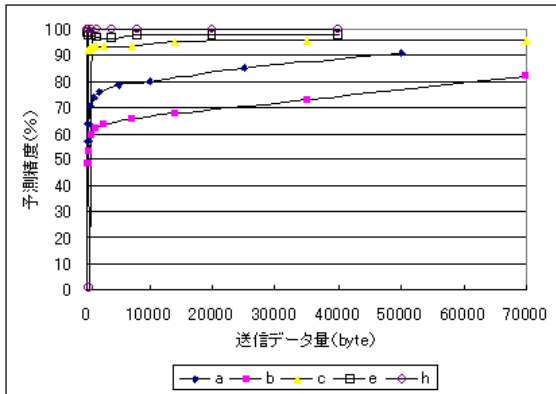


図 8: 送信データ量に対する予測精度

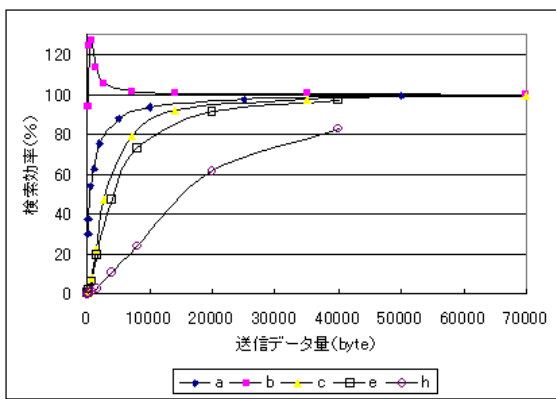


図 9: 送信データ量に対する検索効率

できたということは、利用したデータである人力車の移動軌跡データは速度の変動が少ないオブジェクトだったということがわかる。

尚、最大速度を用いた予測手法により、きわめて 100% に近い予測精度を得ることができたが、その分検索効率は悪くなった。この中で、最大速度のリセット、グルーピング、四方向それぞれで最大を求めるなどの手法を導入した手法 c が、それほど予測精度を落とさずに検索効率を向上させるという結果を得た。

5 まとめ

本研究では、分散移動軌跡データベースのための予測型インデックス手法を提案した。そして、それらを比較考察するための分散型移動軌跡データベースを実際に作成し、性能の比較評価を行った。これにより、予測型インデックスにおけるグルーピング手法の導入が検索効率を向上させ、最大速度に基づいて予測 MBR の大きさを決め

る手法が予測精度を向上させることが分かった。

今後の課題としては、グルーピング手法の改良がある。本研究では、物体の初期位置のみを考慮したグルーピングを行ったが、初期位置が近くてもそれ以降の移動軌跡が必ずしも近くない場合がある。この場合は、逆に検索効率が悪くなる可能性もある。例えば速度情報を考慮するグルーピングにより、移動方向の同じオブジェクトをグルーピングでき、さらに検索効率のよいインデックスが作成できると考えられる。また、今回は人力車の移動軌跡データを用いたのみであったが、それ以外の対象物についても予測手法の評価実験を進めることを検討している。

謝辞

本研究を進めるにあたって多大なご協力を頂いた大阪大学サイバーメディアセンターの村田正幸教授には、この場を借りて厚く御礼申し上げます。

参考文献

- [1] Swarup Acharya, Viswanath Poosala, and Sridhar Ramaswamy. Selectivity estimation in spatial databases. pp. 13–24, 1999.
- [2] I. F. Akyildiz, Y. Sankarasubramaniam W. Su, and E. Cырci. Wireless sensor networks: a survey. *Computer Networks*, 2002.
- [3] E. Clementini and P. Di Felice. Topological invariants for lines. *IEEE Transaction on Knowledge and Data Engineering*, Vol. 10, No. 1, pp. 38–54, 1998.
- [4] O. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD'84 Conference Proceedings*, pp. 47–57, 1984.
- [5] Patrick Laube and Stephan Imfeld. Analyzing relative motion within groups of trackable moving point objects. In *GIScience 2002 Conference Proceedings*, pp. 132–144, 2002.
- [6] Yufei Tao, Jimeng Sun, and Dimitris Papadias. Selectivity estimation for predictive spatio-temporal queries. In *Proceedings / International Conference on Data Engineering, Bangalore, India*.
- [7] Michalis Vazirgiannis and Ouri Wolfson. A spatiotemporal model and language for moving objects on road networks. In Christian S. Jensen, Markus Schneider, Bernhard Seeger, and Vassilis J. Tsotras, editors, *SSTD 2001*, Vol. 2121 of *Lecture Notes in Computer Science*, pp. 20–35. Springer-Verlag, 2001.
- [8] 杉野勝敏, 朝倉康夫, 羽藤英二. 移動体位置情報を用いた交通行動シミュレーションモデルの開発. 土木学会, 第 26 回土木計画学研究発表会 (講演番号 188), 2002.
- [9] 柳沢豊, 赤埴淳一, 佐藤哲司. 移動軌跡データに対する類似度検索手法. FIT2002 一般講演論文集第二分冊, pp. 37–38, 2002.
- [10] 白石陽, 安西祐一郎. 分散センサデータベースの閲覧のためのインクリメンタルなデータ提供方式. トランザクション「データベース」, 第 44 巻, 2003.