移動軌跡ストリームデータのためのインクリメンタルなヒストグラムの管理手法

An Incremental Histogram Management Method for Moving Trajectory Stream Data

町田 陽二♢ 石川 佳治♡
北川 博之♡

Yoji MACHIDA Yoshiharu ISHIKAWA
Hiroyuki KITAGAWA

GPS や通信技術の発展に伴い，移動する多数のオブジェクトの移動状況の追跡が容易になっている．こうした移動状況データを分析・予測に利用するには，ストリーム的に配信されてくる移動状況データを効率よく要約する必要がある．そこで，我々はマルコフ連鎖モデルに基づき移動データを要約する移動ヒストグラムを動的に構築する手法の開発を進めている．提案手法ではヒストグラムを表現する物理的なデータ構造として木構造を採用し，移動パターンを複数の粒度で表現する．移動オブジェクトの移動軌跡が送られた際には，インクリメンタルにヒストグラムを更新する．また，本論文では，中間ノードにカウンタを設けた場合の処理時間を評価する．

With the recent progress of spatial information technologies and communication technologies, it becomes easy to track trajectories of many moving objects in real-time. To use obtained moving object trajectories for the analysis and prediction, we need to accumulate given trajectory streams in an efficient and accurate manner. For this purpose, we propose a mobility histogram construction method based on the Markov chain model. The histogram is physically represented as a tree structure and represents movement patterns in multiple granularities. When a new trajectory sequence is obtained, it updates the histogram structure incrementally. We also evaluate processing time for a method in which middle-nodes contain counters.

1. はじめに

GPS や通信技術の発展に伴い，移動する多数のオブジェクトの移動状況の追跡が容易になっている．また，こうしたオブジェクトの移動状況を蓄積するのに，時空間データベースがますます一般的になっている．

大量の移動オブジェクトの移動状況をリアルタイムに追跡し，分析・予測に役立てるには，ストリーム的に配信されてくる移動状況データを効率よく要約することが求められる．さらに移動オブジェクトの移動状況を要約することは，オブジェクトの移動状況を蓄積した時空間データベースにおける問合せ処理でも有用である．

♢ 学生会員 筑波大学理工学研究科
y-machida@kde.cs.tsukuba.ac.jp

♡ 正会員 筑波大学システム情報工学研究科，計算科学研究センター
{ishikawa,kitagawa}@cs.tsukuba.ac.jp

ストリーム配信されるデータを継続的に集計し，コンパクトに移動状況を要約することは重要な研究課題である．そこで本研究グループは，移動データを要約する**移動ヒストグラム**（mobility histogram）の概念を提案し，ストリーム的に配信されてくる移動状況データを効率よく集計するための動的なヒストグラム構築手法の研究を進めている [11]．このアプローチでは，マルコフ連鎖モデルに基づいて移動パターンの移動統計量を集約する．移動オブジェクトの移動軌跡が送られるたびに，インクリメンタルにヒストグラムの更新を行う．精度を落とさずに効率的な処理が実現できること，また，コンパクトにヒストグラムを表現できることが課題となる．我々のグループの過去の研究 [11] では，木構造に基づくヒストグラム構造のアイデアを提案した．この手法では，移動軌跡データが少ない時点では少数のバケットからなる木構造に移動データを要約して管理する．移動データが追加されるにしたがって木構造を順次拡張し，指定された上限のサイズに達した時点で，必要に応じて木の枝刈りなどを行う．

しかし，このアプローチは効率の面では優れていたものの，精度の面では不十分であった．初期の時点において移動軌跡データを要約して表現してしまうため，そこで発生した誤差が後々の段階まで波及してしまうという点がその理由であった．

そこで，本研究では，ヒストグラムに対する新たなデータ構造の提案を行う．提案手法では，初期段階からメモリを積極的に利用し，移動オブジェクトの移動状況を正確に反映することを目指す．加えて [6] では，**忘却** (forgetting) の概念の導入によるヒストグラム構造の拡張を図った．忘却の概念により，時間につれて移動パターンが変化するような非定常的な移動状況への対応を行う手法である．また，忘却の概念を用いた木構造のヒストグラムのメンテナンス処理において，メモリサイズの制限も考慮することにより，コンパクトなデータ構造の表現を試みた．本稿では，定常状態で中間ノードにカウンタを設けた場合の処理時間の評価を示す．提案手法の一部である忘却の概念を取り入れた手法についての説明は割愛する．詳細については [6] を参照していただきたい．

2. 背景および関連研究

ヒストグラムは，データを要約するための一手法であり，データベースの分野でも，特に問合せ最適化や近似問合せにおいて盛んに研究開発が進められている [5]．これまで特に単一属性に関するヒストグラム構築の研究が進められてきたが，近年では複数属性に対するヒストグラム構築に関する技術開発も着目されている [2, 8]．また，動的なヒストグラムの管理に関しても研究が進められている [4, 7]．

本研究で対象とするヒストグラムは，特殊な制約はあるが多次元のヒストグラムと分類される．与えられたデータに対して最適なヒストグラムを構築することは NP 完全に属する困難な問題であり [5]，近似的な構築法が求められる．また，本研究の目的とする移動軌跡データのリアルタイムの集積を実現するためには，インクリメンタルな更新に対応するために，特に効率性が求められる．

移動状況データをヒストグラム表現することで，大量の移動オブジェクトの移動状況を効率よく分析予測することが可能になる．例えば，移動オブジェクトがどこにいた可能性が高いのか，移動オブジェクトが今後どこに行く可能性が高いのか，ある領域においてオブジェクトがどのように移動するのかなどは統計量をもとに推定できる．関連研究 [3, 10] と比較した本研究の他の特徴としては，ストリーム的な移動軌跡データのリアルタイム処理がある．静的なデータを集約するのではなく，大量のストリームデータを効率的に集約する点を特徴とする．

3. 移動パターンのモデリング

3.1 Z-ordering

まず，準備として，本手法における移動パターンのモデリングのために用いる，2 次元平面の番号付け手法について述べる．この手法は 2 次元平面を 1 次元で順序付けする手法の一つである **Z-ordering**[9] に基づいている．複数ある順序付けする手法の中でも，Z-ordering には順序付けの計算が容易でかつ，様々な分割にも扱えるというメリットがあるので，本研究でも利用している．

2 次元空間が各次元ごとに 2^P（P は分割レベルを表す）個ずつ，$C = 2^{2P}$ 個のセルに分割されているとする．この分割のことを**レ**

ベル P の分割と呼ぶ．各セルには，2P ビットのセル番号を付与する．セル番号の上付き数字で，空間分割レベルを表す．

このレベルの概念を用いることで空間分割の粒度を指定できる．レベルの異なる分割を対比して，粗い分割の方を上位の空間分割，細かい分割の方を下位の空間分割と呼ぶ．

3.2 マルコフ連鎖モデル

時空間データ分析における**マルコフ連鎖モデル**（Markov chain model）は，ある地域から別の地域へある期間内にどの程度の人口が移動したなどの，移動オブジェクトの時空間的な移動傾向の把握に用いられる [12]．以下では，マルコフ連鎖モデルの概念について述べる．

図 1 は，上位空間と下位空間の場合について時刻 $t = \tau$ でそれぞれセル $2^{(1)}$，セル $9^{(2)}$ にいたオブジェクトが，次の時刻 $t = \tau + 1$ でセル $3^{(1)}$，セル $12^{(2)}$ に，そして $t = \tau + 2$ の時点でセル $1^{(1)}$，セル $6^{(2)}$ に移動した状況を示している．

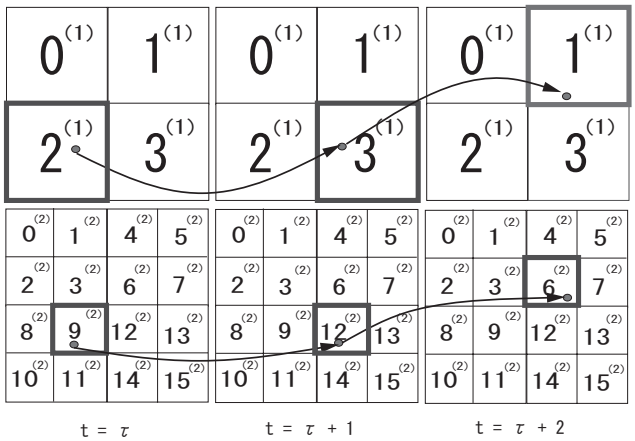

図 1: マルコフ連鎖モデルの概念（2 次の場合）
Fig. 1: Concept of Markov chain model (when n = 2)

図 1 の $9^{(2)} \rightarrow 12^{(2)} \rightarrow 6^{(2)}$ という移動の例は 2 次のマルコフ遷移である．このような遷移情報を集積しておけば，「$9^{(2)} \rightarrow 12^{(2)}$ と遷移したオブジェクトが次にどのセルに移動する確率が高いか」，「$12^{(2)} \rightarrow 6^{(2)}$ と移動したオブジェクトは，セル $12^{(2)}$ の前にどこにいた可能性が高いか」といった質問に，確率の推定値をもとに答えることができる．また，ある領域においてオブジェクトがどのように移動するかといった移動パターンの分析などにも利用できる．

本稿で提案するヒストグラムは，指定された遷移の次数 n に対し，n 次のマルコフ遷移を表現する．ヒストグラムの構造について，次節で説明する．なお，以下の説明では n = 2 の場合を想定するが，アルゴリズムは他の n の値にも適用可能である．

4. 提案手法

4.1 データ構造

4.1.1 基本的アイデア

入力として与えられる移動軌跡データにおいて，分割レベル m でセルの番号付けがなされているとする．m を**最大空間分割レベル**と呼ぶ．移動ヒストグラムは木構造で表現する．木の各ノードは 0 個以上 4 個以下の子を有し，そのノードに対応する移動軌跡の数を集積するためのカウントも有する．

ここで，遷移シーケンス $a^{(m)} \rightarrow b^{(m)} \rightarrow c^{(m)}$ を挿入することを想定する．$a^{(m)}$，$b^{(m)}$，$c^{(m)}$ はセル番号を表す．また，$a^{(m)}$ をステップ 0 のセル，$b^{(m)}$ をステップ 1 のセル，$c^{(m)}$ をステップ 2 のセルと呼ぶ．上記遷移シーケンスを木に挿入する場合を例にとって説明する．まず $a^{(m)}$，$b^{(m)}$，$c^{(m)}$ をバイナリ表記して，まず，$a^{(m)}$ の上位 2 ビットを取り出す．その内容が $00\,(= 0^{(1)})$，$01\,(= 1^{(1)})$，$10\,(= 2^{(1)})$，$11\,(= 3^{(1)})$ のいずれかであるかに応じて，対応する辺を辿り，子ノードへと達する．ただし，そのような辺がない場合には，新たに辺を作成する．さらに $b^{(m)}$，$c^{(m)}$ の順に上位 2 ビットを取り出し，その値に応じて同様に子ノードにアクセスする．以上のステップは空間分割レベル 1 に対応する．

次に，$a^{(m)}$ の次の 2 ビット，$b^{(m)}$ の次の 2 ビット，$c^{(m)}$ の次の 2 ビットをそれぞれ取り出し同様の処理を行う．このステップは空間分割レベル 2 に対応する．このようなステップを繰り返し，2m ビットずつを処理した時点で末端ノードに至る．この末端ノードまで至る過程において，各ノードのカウントを 1 ずつインクリメントする．図 2 に m = 2 の場合について，$3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$ を追加する例を示す．

図 2 の点線はノードを辿るような遷移シーケンスが依然として到着していないことを表しており，実際にはそれ以下の部分木は存在しない．実線はノードを辿るような遷移シーケンスが過去に挿入されたことを表している．

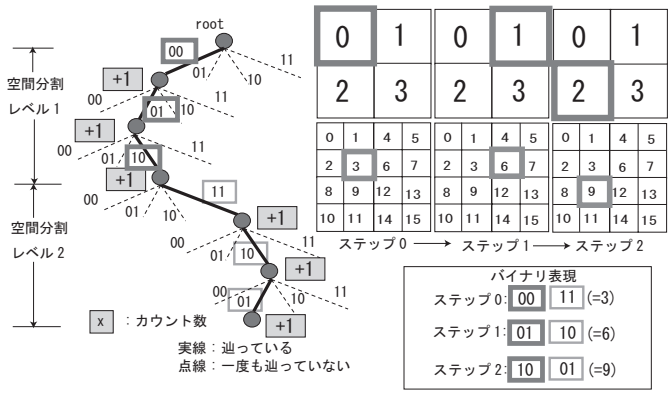

図 2: 木構造 ($3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$，m = 2)
Fig. 2: Tree structure ($3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$，m = 2)

4.1.2 追加アルゴリズム

追加処理のアルゴリズムでは，最大空間分割レベルが m であり，マルコフ遷移の次数が n であることを想定している．ヒストグラムの木構造の各ノードは，カウント値と子ノードへの 4 個のポインタを有する．初期段階では，木構造はルートノードただ 1 つからなる．

アルゴリズムの概要を説明する．まず，ノードへのポインタ node を，ヒストグラムのルートである root で初期化する．次に，シフト量 shift を初期化する．続いて粗い分割から細かい分割へ辿るのに，分割粒度を用いる．各分割粒度ごとに 0 番目のステップから n 番目のステップという遷移シーケンスの遷移を進める．ノードを辿る過程でまだ割り当てられていないノードが存在した場合，ノードを新たに割り当てる．このような処理により，1 つの遷移シーケンスの追加において辿る辺の数は $m \times (n + 1)$ 個となる．ヒストグラムの構造は主記憶上に置かれ，また，m, n の値は一般に小さいため，追加処理は高速に行える．

ここでヒストグラムのサイズについて考える．すべての移動パターンについて遷移シーケンスが木構造に挿入されることを考えると，最大 $4^{m \times (n+1)}$ のノードが必要になる．各ノードにおいて，カウンタが 4 バイト整数であり，4 個のポインタが 4 バイトであるとすると，各ノードのサイズは 20 バイトとなる．そのため，たとえば，m = 4，n = 2 の場合は最悪のヒストグラムのサイズは 320MB となる．一方，m = 5，n = 2 と一段詳細度を上げると，最悪のサイズは 20GB となり膨大となる．

ただし，実際の移動オブジェクトの移動パターンには大幅な偏りが存在する．ある時点である位置に存在するオブジェクトが，次の時点で大きく離れた地点に移動することは現実的にはありえないことから，たとえば図 1 において $0^{(2)} \rightarrow 15^{(2)} \rightarrow 0^{(2)}$ という遷移シーケンスは発生しえない．このため，実際の利用においては木構造のほんの一部のみが実体化されるため，実際に要求されるメモリ領域は上記よりは大幅に小さいと考えられる．

ヒストグラムの木構造に関しては，実装レベルで配列などを用いてよりコンパクトに表現することが考えられる．しかし，上記のアプローチは基本的にデータを追加していくだけであるため，データの追加に伴い木が巨大になる可能性は避けられない．また，過去の遷移シーケンスと最近の遷移シーケンスのカウントが全く同等の扱いになっているので，時間的に移動パターンが変化するような非定常的な状況に適用できない．[6] では，こうした問題に

対し忘却の概念を導入することで解決を図っている．

4.2 問合せ処理

次いで，ヒストグラムを用いて，移動軌跡のカウント数を求める問合せに答えるための問合せ処理方式について述べる．

4.2.1 基本的アイデア

まず，最大空間分割レベルが m = 2 の場合について，例を用いて処理の概要を説明する．

1. 移動軌跡の各セルのレベルが最大空間分割レベルと一致する場合：たとえば，$3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$ を考える．この場合が一般的で容易である．遷移シーケンスの挿入と同様に，各ステップをバイナリ表記して各レベルごとに 2 ビットずつ辿っていき，目的のカウントを返す．
2. 移動軌跡の各セルのレベルが最大空間分割レベルよりも粗い場合：例として $1^{(1)} \rightarrow 1^{(1)} \rightarrow 9^{(2)}$ を考える．(1) と同様に，各ステップをバイナリ表現にして，各レベルごとに 2 ビットずつ辿っていく．この場合は，空間分割レベル 1 の $01 \rightarrow 01 \rightarrow 10$ の後，$\{00,01,10,11\} \rightarrow \{00,01,10,11\} \rightarrow 01$ の $4 \times 4 \times 1 = 16$ 通りの経路をすべて辿り各カウントを調べ，総和をとる必要がある．
3. 移動軌跡のセルのレベルが最大空間分割レベルよりも大きい場合：たとえば，$3^{(2)} \rightarrow 25^{(3)} \rightarrow 9^{(2)}$ を考える．$3^{(2)} \rightarrow 25^{(3)} \rightarrow 9^{(2)}$ は，空間分割レベル 2 までの木では表現できないので，カウントを近似的に推定する必要がある．実際はカウントの推定式を定め，近似することが望ましいが，本稿では便宜上このような問合せに対して，カウント 0 を返すことにする．

4.2.2 問合せ処理のアルゴリズム

問合せ処理のアルゴリズムを**アルゴリズム 1** に示す．このアルゴリズムは再帰的な関数 est_count() を定義している．この関数は 4 つの引数 node, qseq, levels, step をとる．node は処理対象のノードへのポインタ，step は現在処理対象の遷移ステップを表す．qseq, levels はそれぞれ問合せ移動シーケンス中の領域番号の配列およびレベル番号の配列である．たとえば，$3^{(1)} \rightarrow 14^{(2)} \rightarrow 15^{(2)}$ という遷移シーケンスが問合せとして与えられた場合，qseq[0] = 3, qseq[1] = 14, qseq[2] = 15 であり，levels[0] = 1, levels[1] = levels[2] = 2 となる．

ある遷移シーケンスの出現回数の推定を求める問合せがユーザから与えられると，まず 2 つの配列 qseq と levels を構築し，est_count(root, qseq, levels, 0) という関数呼び出しを行う．root はヒストグラムのルートを表し，0 はまず最初は 0 番目の遷移ステップに関する処理を行うことを意味する．

Algorithm 1 est_count(node, qseq, levels, step)

```
1:  input  node: 対象ノード, qseq: 問合せの遷移シーケンス, qlevel: 各遷移のレベル配列, step: 処理対象のステップ
2:  if (i = 0,...,n について levels[i] ≡ 0)
3:    { 問合せ処理を終え，目的のカウントを返す }
4:    if (levels[step] > 0)
5:      shift := 2(levels[step] − 1); //2 ビット抽出のシフト量
6:      c := { 探索する子ノード番号 }
7:      levels[step]−−;
8:      if (node.child[c] ≡ NULL) //探索途中で辿る辺がない
9:        return count := 0;
10:     else
11:       { 次の遷移ステップを表す子ノードへ再帰処理 }
12:   else
13:     {4 つの子ノードに分岐し，計算結果を集計 }
```

アルゴリズムの概要を説明する．このカウント処理では，ルートから 1 ノードずつ辺を辿って，対象となるカウントを有するノードまで探索し，そのカウント値を返す．木の探索処理は再帰呼出しで実現する．ただし，場合によっては 1 つの問合せに対し複数の辺に分岐して探索を行う必要も生じることに注意する．

- 4 行目：与えられた問合せをどこまで処理したかを表現するのに levels 配列の値で判別
- 4-11 行目：node からその子に対し 1 つだけ辺を辿る処理
  - 8 行目の条件が偽：11 行目の再帰呼び出しを実行
    ⇒ 子に対してカウント処理を引き継ぐことを表す
  - 8 行目の条件が真：4.2.1 節の 3 の場合のように，問合せに対してヒストグラムの解像度が高くなく，近似的なカウント値を推定する必要がある場合の処理
    ⇒ ここではカウント 0 を返す
- 13 行目：分岐処理が必要となり，4.2.1 節の 2 の場合に相当
  ⇒ 4 つの辺に対し，est_count() を 4 回再帰呼び出しし，その結果の和をとる

5. 実験

忘却を持たない定常状態のときで，中間のノードにカウンタを設ける場合の処理時間について評価を行う．

ヒストグラムの処理時間についてそれぞれ挿入処理，問合せ処理における様子を見ていく．

5.1 実験データ

実験では Brinkoff により作成された移動オブジェクトデータ生成ソフトウェアを用いる [1]．これは実際の市街地の道路ネットワーク上を自動車などが移動する際の移動の状況をシミュレーションするシステムであり，これによって生成された移動データを実験において使用する．今回扱うデータは，このシステムで提供されているドイツ Oldenburg 市の市の中心部（約 2.5×2.8km）の道路ネットワークにおいて，パラメータを適宜設定して生成したものである．図 3 にこのシステムのインタフェース画面を示す．

以下の実験において，Pentium4 3.2GHz の CPU， 1 GB のメモリを持つ PC を使用して実験を行った．加えて地図領域を 1024×1024 のセル領域に分割した場合を想定している．

5.2 実験手法

移動オブジェクトデータ生成ソフトウェアより，二つのデータセット（A），(B）を用いてヒストグラムの構築時間，また問合せ処理時間の実験を試みた．データセット (A) は 2 次のマルコフ遷移をし，それぞれ空間分割レベル 10 の遷移シーケンスを 1000 件有する．データセット (B) はデータセット (A) 同様の遷移シーケンスを 10000 件有する．

5.2.1 ヒストグラム構築時間

定常状態の挿入処理を行う．そこでデータセット（A）でヒストグラムを構築すると，ノードが 25549 個生成される．各ノードには，カウンタに 4 バイト，各ポインタに 4 バイト，計 20 バイト使用している．これより，上記のデータセットで構築したヒストグラムのサイズには 500 kバイト必要となる．ヒストグラムの構築時間は 19 ミリ秒である．遷移シーケンスあたりの処理時間は 0.019 ミリ秒である．同様にデータセット (B) ではノードが 191136 個生成される．構築したヒストグラムのサイズには 4M バイト必要となる．ヒストグラムの構築時間は 150 ミリ秒である．遷移シーケンスあたりの処理時間は 0.015 ミリ秒である．各データセットのヒストグラム構築時間を図 5 に示す．

5.2.2 問合せ処理時間

定常状態の問合せ処理を行う．複数の問合せパターン 100 個を与えた際の問合せ処理時間を調べた．問合せのパターンには 4.2.1 節のように (1) 移動軌跡の各セルのレベルが最大空間分割レベルと一致する場合（問合せ例：$177800^{(10)} \rightarrow 176121^{(10)} \rightarrow 526585^{(10)}$），(2) 移動軌跡の各セルのレベルが最大空間分割レベルこの場合よりも粗い場合（問合せ例：$3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$，$53662^{(9)} \rightarrow 66816^{(9)} \rightarrow 109748^{(10)}$) を用いた．

$3^{(2)} \rightarrow 6^{(2)} \rightarrow 9^{(2)}$ を問合せ例として分割された地図領域を図 4 に示す．太枠は空間分割レベル 2 を，細枠は空間分割レベル 4 を表している．

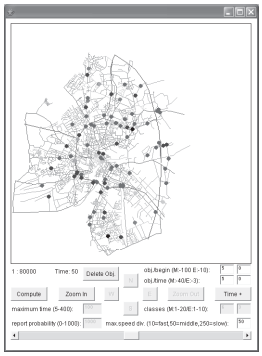

図 3: 移動軌跡データの例
Fig. 3: Moving trajectory data

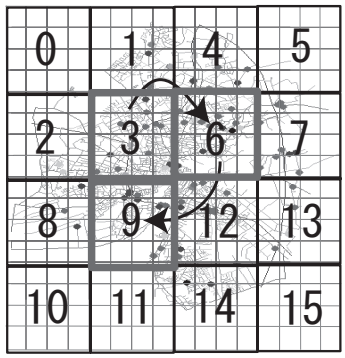

図 4: 地図領域における問合せ例
Fig. 4: Example query

いずれの問合せも全て 2 次のマルコフ遷移の遷移シーケンスで行った．問合せパターンごとの問合せ処理時間を図 6 に示す．

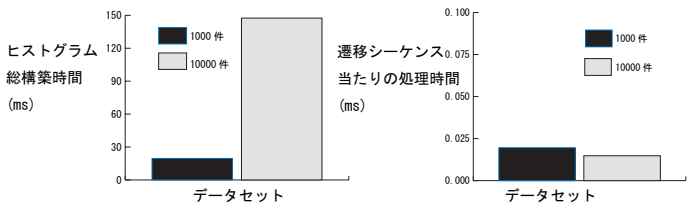

図 5: ヒストグラムの構築時間
Fig. 5: Histogram construction time

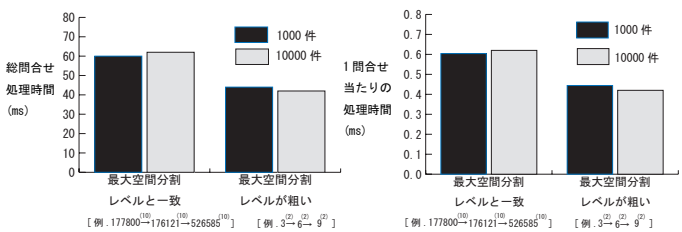

図 6: 問合せ処理時間
Fig. 6: Query processing time

データセット (A) で構築したヒストグラムに各問合せパターンの平均処理時間は，(1)60 ミリ秒，(2)44 ミリ秒である．1 問合せ当たりの処理時間は (1)0.60 ミリ秒，(2)0.44 ミリ秒である．1 問合せあたりの問合せ結果は (1)0.36 回，(2)26 回のカウントである．同様にデータセット (B) では，(1)62 ミリ秒，(2)42 ミリ秒である．1 問合せ当たりの処理時間は (1)0.62 ミリ秒，(2)0.42 ミリ秒である．1 問合せあたりの問合せ結果は (1)0.72 回，(2)173 回のカウントである．

実験結果より，中間ノードにカウンタを有する場合のヒストグラム構築時間はデータセットが 10 倍に増えても処理時間が 10 倍にはならなかった．それはデータ数が多ければ新規にノードを作成することも減り，上位レベルのノードを辿る可能性が高いと考えられ，処理時間が短縮している．また，問合せ処理時間においては，データセットが多くなっても処理時間に差がほとんどない．すなわち同じ空間分割レベルでヒストグラムを構築していれば，問合せ処理に影響を及ぼさないことがいえる．

6. まとめ

本稿では，木構造による移動統計量の推定手法を提案し，定常状態において，中間のノードにカウンタを有する場合の処理時間について評価した．

今後の課題として，1) 末端のノードのみカウンタを有する場合の処理時間についての同様な評価実験，2) 定常的な場合と非定常的な場合の性能評価，3) メモリの回収具合を考慮した実装方式の検討が挙げられる．

[謝辞]

本研究の一部は，日本学術振興会科学研究費基盤研究 (C)(2)(16500048)，旭硝子財団研究助成，稲森財団研究助成，文部科学省科学研究費特定領域研究 (2)(16016205) 及び CREST「自律連合型基盤システムの構築」による．

[文献]

[1] T. Brinkhoff. A framework for generating network based moving objects. *GeoInfomatica*, No. 6(2), pp. 153–180, 2002.

[2] N. Bruno, L. Gravano, and S. Chaudhuri. STHoles: A multidimensional workload-aware histogram. In *Proc. SIGMOD*, 2001.

[3] Y. Choi and C. Chung. Selectivity estimation for spatio-temporal queries to moving objects. In *Proc. ACM SIGMOD*, pp. 440–451, 2002.

[4] P. Gibbons, Y. Matias, and V. Poosala. Fast incremental maintenance of approximate histograms. In *Proc. VLDB*, pp. 466–475, 1997.

[5] Yannis Ioannidis. The history of histograms (abridged). In *Proc. VLDB*, 2004.

[6] 町田陽二, 石川佳治, 北川博之. 移動軌跡ストリームデータのための動的ヒストグラム構築手法. 電子情報通信学会データ工学ワークショップ (DEWS 2005), March 2005.

[7] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *Proc. VLDB*, pp. 101–111, 2000.

[8] V. Poosala and Y. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *Proc. VLDB*, pp. 486–495, 1997.

[9] Shanshi Shekhar and Sanjay Chawla. *Spatial Databases*. Prentice Hall, 2002.

[10] Y. Tao, J. Sun, and D. Papadias. Selectivity estimation for predictive spatio-temporal queries. In *Proc. ICDE*, 2003.

[11] 塚本祐一, 石川佳治, 北川博之. 移動軌跡ストリームからの移動統計量推定のための動的ヒストグラム構築手法について. 電子情報通信学会データ工学ワークショップ (DEWS 2004), March 2004.

[12] G. J. G. Upton and B. Fingleton. *Spatial Data Analysis by Example, Volume II: Categorical and Directional Data*. John Wiley & Sons, 1989.

町田 陽二　Yoji MACHIDA
筑波大学大学院理工学研究科在学中．時空間データベースの研究に従事．日本データベース学会学生会員．

石川 佳治　Yoshiharu ISHIKAWA
筑波大学大学院システム情報工学研究科，計算科学研究センター助教授．データベース，データ工学，情報検索などに興味を持つ．日本データベース学会，情報処理学会，電子情報通信学会，人工知能学会，ACM，IEEE CS 各会員．

北川 博之　Hiroyuki KITAGAWA
筑波大学大学院システム情報工学研究科，計算科学研究センター教授．異種情報源統合，文書データベース，WWW の高度利用等の研究に従事．著者「データベースシステム」(昭晃堂) 等．日本データベース学会，情報処理学会，電子情報通信学会，日本ソフトウェア科学会，ACM，IEEE CS 各会員．