マルコフ連鎖モデルに基づく移動ヒストグラムの動的構築法

石川　佳治[†a)]　　町田　陽二[††*]　　北川　博之[†††,††††]

A Dynamic Mobility Histogram Construction Method Based on Markov Chains

Yoshiharu ISHIKAWA[†a)], Yoji MACHIDA[††*], and Hiroyuki KITAGAWA[†††,††††]

あらまし　GPS や通信機器の普及に伴い，移動オブジェクトの追跡が容易となっている．大量の移動オブジェクトの移動状況をリアルタイムにモニタリングし集積することにより，移動状況の把握や予測に利用することが考えられる．そのため，本論文では大量の移動オブジェクトの移動軌跡データをコンパクトに要約するための移動ヒストグラムの構築手法を提案する．ヒストグラムの基礎となる移動統計量のモデルとして，本手法ではマルコフ連鎖モデルを採用し，データキューブに似た移動ヒストグラムの論理表現を与え，OLAP 的な分析を可能とする．一方，その物理表現としては木構造を採用し，近似的な表現により記憶容量の削減を図る．また，移動状況データはストリームとして逐次受信されることから，リアルタイム処理に適した効率的なヒストグラムの構築方式を実現する．本論文では，提案手法の効率及び精度について実験に基づく評価についても報告する．

キーワード　移動オブジェクト，移動分析，ヒストグラム，ストリーム処理，マルコフ連鎖モデル

1. まえがき

無線ネットワーク技術や GPS などの位置測位技術の普及により，今日では自動車などの移動オブジェクト（moving object）[1] の追跡が容易になってきている．多数の移動オブジェクトの移動状況データを集積することにより，移動状況の現状の分析や予測を行うことが可能となると考えられる．

移動状況の分析は，移動オブジェクトの移動軌跡などの情報を蓄積管理する時空間データベースにおける問合せ処理においても重要である [1]〜[3]．特に，時空間データベースにおける選択率（selectivity）の推定に関しては，既にいくつかのアプローチが提案されている [4]〜[6]．

本論文では，大量の移動オブジェクトの移動軌跡を集約する移動ヒストグラム（mobility histogram）の構築手法を提案する．移動パターンを表現するためのモデルとして，本手法ではマルコフ連鎖モデル（Markov chain model）を採用する．マルコフ連鎖モデルは時空間データの解析のための古典的なモデルの一つであり，空間上のオブジェクトの移動がマルコフ連鎖に基づくという仮定に基づいている [7]．このモデルは，交通分析や人口の時間的な流動など，様々な方面で活用されている．

本手法では，移動ヒストグラムの論理表現としてデータキューブ（data cube）[8] を採用し，ユーザに対し OLAP 的な移動分析の機能を提供する．また，リアルタイムの分析処理を支援するため，ストリーム的に配信される移動状況を継続的に集約する，ヒストグラムの動的構築を実現する．ただし，論理的なデータキューブを直接的に実現すると膨大な記憶容量を必要とすることから，木構造に基づくヒストグラムの物理表現方式を提案する．更に，記憶容量の制限のもとで精度の高いヒストグラムを構築するため，取得されるデータに応じた適応的な木構造の伸張を行う，近似的な移動ヒストグラム構築法を示す．適応的な木の伸張の判定のためには統計的な指標を用いるが，ストリー

† 名古屋大学情報連携基盤センター，名古屋市
Information Technology Center, Nagoya University, Nagoya-shi, 464–8601 Japan
†† 筑波大学大学院理工学研究科，つくば市
Master's Program in Science and Engineering, University of Tsukuba, Tsukuba-shi, 305–8573 Japan
††† 筑波大学大学院システム情報工学研究科，つくば市
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305–8573 Japan
†††† 筑波大学計算科学研究センター，つくば市
Center for Computational Sciences, University of Tsukuba, Tsukuba-shi, 305–8577 Japan
* 現在，（株）日立情報システムズ
a) E-mail: ishikawa@itc.nagoya-u.ac.jp

ム的な移動状況データの配信に適した効率的な判定手法を提案する．効率と精度のトレードオフのもとで，有効な移動ヒストグラム構築手法を実現することが本研究の目的となる．

本論文の構成は以下のようになる．**2.** で関連研究について述べる．**3.** では，マルコフ連鎖モデルに基づく移動統計量の概念を導入する．**4.** では移動ヒストグラムの論理表現を示し，**5.** ではその物理表現と，構築・問合せ処理手法について述べる．**6.** では実験及びその結果について述べる．**7.** ではまとめと今後の課題について述べる．

なお，本論文は [9] の内容を拡張したものである．統計処理の部分に関する詳細化を行い，実験とその分析についても整理と拡充を行っている．

2. 関 連 研 究

2.1 時空間データベースにおける集約処理

時空間データベースにおける集約処理について，近年多くの興味がもたれている．[10] では，空間データベース，時制データベース，及び時空間データベースにおける集約問合せの処理方式を広範囲にサーベイしている．ただし，移動オブジェクトに関する議論は含まれていない．

データベースの問合せ最適化においては，しばしば選択率などの統計情報の推測が求められる．空間データベースに関しては，選択率推定に関するアプローチがいくつか提案されている（例：[11]）．しかし，時空間データベースに対してはわずかな提案のみしかない．[4], [5] では，移動する点オブジェクトを管理する時空間データベースにおける空間問合せについて，選択率推定のアプローチを提案している．一方，[6] では，時間に依存して問合せ範囲が変わる時空間問合せに対する選択率の計算法を提案している．これらの選択率推定とは異なり，本手法は大量の移動オブジェクトの移動状況を集約することに焦点を当てている．集約された情報を，移動オブジェクトの移動状況の把握や今後の移動傾向の予測へ利用することを目的とする．

[12] では，移動オブジェクトの移動状況を表す多次元ヒストグラムをインクリメンタルに更新するアプローチを提案しており，空間的な問合せに対し近似的に答えることを目的としている．ただし，移動オブジェクトについては線形の移動を想定している．[12] の手法では，インクリメンタルなヒストグラムの構築に加え，履歴問合せのための古いヒストグラムのアーカイブ処理と，予測問合せのためのヒストグラムのスムージング化を行う．

時空間データベースからの統計量の抽出に関して，我々の研究グループでは，マルコフ連鎖に基づく統計量の利用について研究を行っている．[13] では，索引付けされた大量の移動軌跡データに対する移動統計量の問合せを支援するアルゴリズムを開発している．移動オブジェクトの軌跡データが空間索引 R-木により索引付けされていることを想定しており，統計量計算のために R-木を効率的に利用する．これに対し，本論文では移動軌跡のデータがストリーム的に時々刻々と到着する動的な状況を想定しているため，全く異なるアプローチが必要とされる．

2.2 多次元ヒストグラム及び動的ヒストグラム

ヒストグラム（histogram）[14] は，データベース内のデータの分布をコンパクトに近似表現するものであり，問合せ最適化などにおいて重要な役割を果たす．これまで，様々なヒストグラムの構築手法が提案されてきたが，多くのものは一次元の場合を対象としている．空間データを含む多次元データに対してはいくつかのアプローチが存在している（例：[11], [15], [16]）が，最良の多次元のヒストグラムの構築は NP 困難であると知られていることから [17]，ヒューリスティックスの使用による近似的な表現が用いられる．

一方，動的なヒストグラムの構築と管理に関する研究も近年では盛んに進められている [15], [18]〜[20]．それらのうちで，多次元データに対するものとしては [15], [20] があるが，時空間データベースは対象とされていない．

本論文では，動的かつ多次元のヒストグラムを考える．ただし，動的かつ多次元データに対するヒストグラムに対する既存のアプローチとは異なり，ここで扱う移動ヒストグラムは，移動オブジェクトの性質と，移動統計量として採用したマルコフ連鎖モデルの特性を反映して，次元間の相関性が高い多次元データを扱う必要がある．加えて，ユーザによる探索的な移動分析を支援するために，本手法では時空間的な分割における複数の粒度（granularity）を用いたヒストグラムの論理表現を与える．要求に応じて移動ヒストグラムの粒度を変化させることで，ユーザの要求に合った移動パターンの分析が可能となる．

3. マルコフ連鎖モデルに基づく移動統計量

図 1 に示すように，二次元空間が領域に分割され

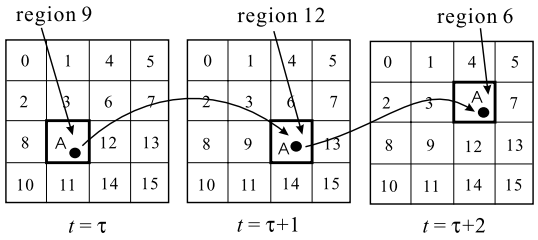

図 1　マルコフ連鎖モデルの概念
Fig. 1　Notion of Markov chain model.

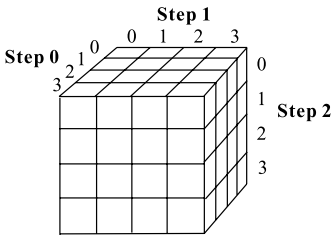

図 2　ヒストグラムの論理表現
Fig. 2　Logical histogram representation.

ているとする．各次元は等しく 2^P に分割されいるため，全体では $R = 2^{2P}$ 個の領域が存在する（図は $P = 2$ の場合）．このような分割のことをレベル P の分割と呼ぶ．各領域に対して $2P$ ビットの番号を Z-ordering 法 [21], [22] に基づいて割り当てる(注1)．図は，時刻 $t = \tau$ に領域 9 にいたオブジェクト A が時刻 $t = \tau + 1$ に領域 12 へ，そして $t = \tau + 2$ では領域 6 に移動したことを表している．

領域 9 にいた別のオブジェクト B が次の時点で領域 12 に移動したとしよう．ここで，B が次に領域 6 に移動する確率を知りたいとする．この確率を Pr(6|9, 12) と表記することにする．領域の間の遷移がマルコフ連鎖モデル（Markov chain model）[7] に従っていると考えると，この確率は 2 次のマルコフ遷移確率（Markov transition probability）ととらえることができる．このような考え方を，更に n 次のマルコフ遷移に拡張することも可能である [9], [13].

上記の確率を推定する単純なアプローチとしては，データベースにすべての移動軌跡を蓄積し，問合せが発生した際にそれらを用いて統計量を計算することが考えられる．この手法は明快であるが，膨大な移動軌跡を格納するため，特に対話的な分析にはオーバヘッドが大となる．そこで，データサイズと計算コストを削減するため，以降では移動状況をコンパクトに表現する移動ヒストグラムの構築法を提案する．

4. ヒストグラムの論理表現

4.1 データキューブの利用

$\langle id, x, y, t \rangle$ の形式のタプルのストリームが入力として得られるとする．id はオブジェクトの ID であり，x, y は時刻 t における座標値である．このようなストリームは，n 次のマルコフ遷移ストリームに容易に変換できる．ただし，パラメータ n はユーザにより指定されるとする．ストリームの各エントリは，例えば $n = 2$ の場合，$9 \to 12 \to 6$ などの n 次の遷移シーケンスとなる．以下では，n 次のマルコフ遷移シーケンスがストリームから連続的に得られると想定する．

n 次のマルコフ連鎖に基づく移動ヒストグラムの論理的な表現として，$n+1$ 次元のデータキューブ [8] を用いる．図 2 は $n = 2$ の例であり，レベル 1 の空間分割（$P = 1$）に対応する．この場合，2 次元空間が $R = 2^{2P} = 4$ 個の領域に分割されているので，データキューブには $R^{n+1} = 64$ 個のセルが含まれる．データキューブの各次元（ステップ 0，1，2 と呼ぶ）は，2 次のマルコフ連鎖の各ステップに対応する．例えば $1 \to 1 \to 2$ という移動シーケンスが得られると，対応するセルの値がインクリメントされることになる．

オブジェクトの移動パターンは必ずしも定常的ではなく，時間とともに移り変わることがある．その場合，長い時間に集積されたデータに基づくヒストグラムは，動的に変化する移動状況を必ずしも適切に表現できない．よって本手法では，ある一定数の連続した遷移シーケンスのグループごとに一つのヒストグラムを構築する．古いヒストグラムはファイルに格納し，後の分析において利用する．このように履歴問合せのために古い統計情報をディスク上に順次出力するアプローチは，[12] にも見られる．

4.2 データキューブを用いた問合せ処理

後述するように，提案手法では移動ヒストグラムの物理的な実装方式として木構造を用いた表現を採用している．しかし，移動分析を行うユーザにとっては，物理構造をイメージして分析することは容易ではなく，また，その必要もない．これに対し，データキューブの論理表現を用いることにより，以下のような利点が

(注1)：Z-ordering 法は多次元データの一次元による番号付けに用いられ，空間的に近い領域に比較的近い番号が割り当てられる．番号の近さと空間的な近さの対応の面では，より好ましい性質をもつ手法が他にも存在するが，Z-ordering 法は計算が容易であり，後述するヒストグラムの効率的な実装において重要な役割を果たす．

生じる．

（1） 分析対象の移動統計情報をとらえることが容易になる．

（2） 分析処理にかかわる操作をデータキューブの概念を用いて表現できる．

（1）については，データキューブのセルを用いて各種統計処理を容易に表現できることが理由となる．例えば，図 2 に示す 2 次のマルコフ連鎖に対するデータキューブがある場合，領域 1 から 2 に移動したオブジェクトが次に 4 に移動する確率を，$\Pr(4|1,\,2) = val(1,\,2,\,4)/val(1,\,2,\,*)$ と計算できる．ここで $val(1,\,2,\,4)$ はセル $(1,\,2,\,4)$ の値であり，$val(1,\,2,\,*) = \sum_{i=0}^{2^P-1} val(1,\,2,\,i)$ である．このような計算は，各セルの値 $val(x, y, z)$ を求める基本演算が支援されていれば容易に実現できる．

更に，例えば「時刻 $t = \tau$ において領域 1 に存在し，時刻 $t = \tau + 2$ に領域 3 に存在したオブジェクトが，時刻 $t = \tau + 1$ において領域 2 に存在する確率を求めよ」といった問合せにも，データキューブ表現は $val(1,\,2,\,3)/val(1,\,*,\,3)$ と利用できる．また，ある時刻においてオブジェクトが密に存在する領域を求める密度問合せ（density query）[23] も，データキューブ上の集約・選択演算の組合せで処理できる．

（2）については，データキューブにおける基本的な演算 [8] を，移動分析においても意味づけ可能である点が理由として挙げられる．例えば，スライス（slice）演算に対応する例としては，図 2 の 2 次の遷移に対するデータキューブにおいて縦方向にセルの総和をとることで，ステップ 0 とステップ 1 からなる 1 次の遷移に対するデータキューブを得ることが考えられる．また，図 2 を用いたダイス（dice）演算に対応する例としては，領域 0, 1 のみの 2 次の遷移に着目して，$2 \times 2 \times 2 = 8$ 個のセルからなるデータキューブの一部を抽出する処理が挙げられる．分析対象のデータの粒度を変化させるドリルダウンとロールアップについては，次節で詳しく述べる．なお，本提案手法においては，これらのデータキューブ演算は仮想的な一種のビューであり，物理的なヒストグラムの構造を変化させるものではない．しかし，移動分析を行うユーザの立場からは，操作がより理解しやすくなることから有用であると考えられる．

ここでデータキューブの次数の設定について述べる．n 次のマルコフ連鎖に対するデータキューブがあれば，1 次から $n-1$ 次の遷移の確率も計算できる．ただし，大きな n に対するデータキューブのサイズは膨大となる．そのため，例えば $n = 4$ 次の遷移パターンの統計を 2 次の遷移パターンの組合せをもとに推定するなどの工夫が実際には必要となる．そこで以下では，主に $n = 2$ 次の遷移について検討する．このような考え方は，自然言語処理で単語の出現パターンを分析する N グラム統計などでもとられる [24].

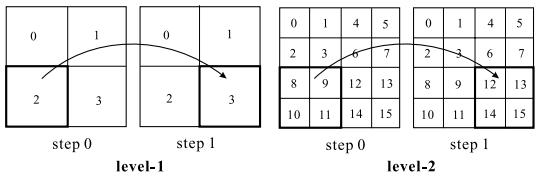

図 3 ロールアップとドリルダウン
Fig. 3 Roll-up and drill-down.

4.3 ロールアップ処理とドリルダウン処理

移動パターンの分析では，移動データを異なる粒度で見たいことがある．例えば，単位時間 30 秒において分析をしていたユーザが，より大まかな分析のためそれを 1 分に変更して分析することが考えられる．これは時間に関するロールアップ（roll-up）操作 [8] に対応する．逆に，より大きい単位時間の使用はドリルダウン（drill-down）操作に対応する．これらの操作を支援するため，本手法ではユーザが指定した異なる単位時間ごとに複数のデータキューブを構築する．ヒストグラムの構築・格納コストが小さければ，複数のヒストグラムの構築は深刻な問題ではないといえる．

図 3 をもとに空間的なロールアップとドリルダウンの操作について述べる．ここでは一次の遷移を考える．左の図はレベル 1 の空間分割，右の図はレベル 2 の空間分割を表す．レベル 1 の分割をもとに構築されたデータキューブを，レベル 2 のデータキューブをロールアップしたものととらえ，その逆をドリルダウンとみる．次章で述べるように，本手法では異なる空間の分割に対し複数のヒストグラムを構築することはせず，単一のヒストグラムで対応する．物理的にはヒストグラムを木構造で表現するが，異なる空間分割への対応は木の適応的な伸張により実現する．

5. ヒストグラムの物理表現

論理的な移動ヒストグラムの直接的な実装には多大なオーバヘッドが存在する．前述のように領域の総数は $R = 2^{2P}$ であり，n 次のマルコフ連鎖に対するデータキューブは R^{n+1} 個のセルを有する．例えば

$P = 5$，$n = 2$ の場合 $R^{n+1} = 1024^3$ であり，約 100 億個のセル数となる．各セルを 2 バイトで表現したとしても 2 GByte のメモリが必要となる．ヒストグラムのサイズの削減と動的な維持管理を可能とするため，以下では木構造に基づく物理表現を提案する．

5.1 ヒストグラムの構造とその構築

5.1.1 基本構造（素朴な方式）

物理的なヒストグラムは，トライ（trie）[25]〜[27] に似たデータ構造を有する．ただし，一般のトライに対し以下のような相違点がある．

（1） 通常のトライは，キーの各ビットが 0 か 1 かによって左右に分岐する二分木の構造をもつ．これに対し本手法は，2 次元空間を縦横に分割した四つの領域にそれぞれ対応する，00，01，10，11 の 4 通りの分岐を行う．これにより，quadtrie [25] あるいは quadtree [28] のような空間分割を行う．

（2） 木のルートからの最初の分岐はマルコフ連鎖のステップ 0 について，次の分岐はステップ 1 について，というように，各分岐ごとに順次遷移のステップを対応づける．このような考え方は，多次元トライ（multidimensional trie）[26], [29] や k-d 木（k-d tree）[28] に関連している．

（3） 一般にトライはキー値の格納と探索に用いられるが，本データ構造はカウント値の保持とその利用のために用いられる．

レベル M の分割による入力遷移シーケンス（例：$2^{(M)} \to 10^{(M)} \to 12^{(M)}$）が与えられたとする．$r^{(M)}$ という記法は，領域 r の分割レベルを明記する場合に用いる．レベル M より細かいレベルで統計情報を表現できないことから，M を最大分割レベルと呼ぶ．

木の各ノードは 0 から 4 個までの子ノードを有し，各ノードには対応する遷移シーケンスの出現回数を表すカウンタをもたせる．例として，2 次の遷移シーケンス $r_0^{(M)} \to r_1^{(M)} \to r_2^{(M)}$ が与えられたと考える．ただし，$r_0^{(M)}$，$r_1^{(M)}$ 及び $r_2^{(M)}$ は領域の番号である．$r_0^{(M)}$，$r_1^{(M)}$，$r_2^{(M)}$ をそれぞれ，ステップ 0，1，2 の領域と呼ぶ．以下のように処理を進める．

（1） 木のルートから処理を開始する．

（2） 領域番号 $r_0^{(M)}$ をバイナリーで表し，その最初の 2 ビットを抽出する．その値は 00 ($= 0^{(1)}$)，01 ($= 1^{(1)}$)，10 ($= 2^{(1)}$)，11 ($= 3^{(1)}$) のいずれかとなるが，対応する枝をたどって子ノードを訪問する．子ノードが割り当てされていなければ，新たにノードを生成し，親ノード（ルート）からの枝を作成する．

（3） $r_1^{(M)}$ のバイナリー表現から最初の 2 ビットを抽出し，対応する枝をたどる．ノードが割り当てされていない場合はステップ 2 と同様に処理する．

（4） $r_2^{(M)}$ を同様に処理する．以上のステップ 2 から 4 は，レベル 1 の粗い分割の遷移に対応する．

（5） $r_0^{(M)}$，$r_1^{(M)}$，$r_2^{(M)}$ のバイナリー表現のそれぞれから次の 2 ビットの値を抽出し，同様に枝をたどる．これらはレベル 2 の分割に相当する．

（6） すべてのビットが消費されるまでこのような処理を繰り返す．

なお，ノードを訪問した際には，そのカウンタをインクリメントするものとする．各遷移ステップごとに交替で木を伸張させるアプローチは k-d 木に似ており，各空間を四つずつに繰り返し分割するアプローチは四分木に対応する．

図 4 は，$M = 2$ の場合に木に $3^{(2)} \to 6^{(2)} \to 12^{(2)}$ という遷移シーケンスを追加する例を示している．点線の枝は，対応する遷移シーケンスがまだ到着したことがなく，実際には辺が存在しないことを表す．

3. で述べたように，本手法では領域に対して Z-ordering 法による番号付けを行っている．Z-ordering 法を用いる利点は，上述のように 00，01，10，11 の 4 通りに分岐しながら木をたどることが，ちょうど Z-ordering による番号付けに対応することである．例えば，図 4 における分割レベル 2 のステップ 0 の領域 $3^{(2)}$ は，分割レベル 1 では領域 $0^{(1)}$ に対応している．これは，$3^{(2)}$ を 2 進数で表現した 0011 の上位 2 ビットが 00 であることより成り立つ．この Z-ordering 法の性質は，アルゴリズムの簡略化の上で大いに役立っている．

5.1.2 近似的な表現

前述の物理構造は正確であるが，メモリの消費量が大きい．そこで，与えられたノード数の上限値 N のもとでの近似的な表現を考える．基本的なアイデアは，遷移シーケンスの分布を考慮した木の適応的な伸張である．図 5 を用いて説明する．左は，ある時点での 2 次のマルコフ遷移に対するヒストグラムの例である．図中の葉ノード（黒色）について考える．これは，ステップ 0 の最初の 2 ビットが 01 で，ステップ 1 の最初の 2 ビットが 11 であるので，遷移シーケンス $1^{(1)} \to 3^{(1)} \to *$ に対応する．このノードには独自のバッファ領域が付随しており，これまで到着した対応する遷移シーケンスをすべて保持している．

このバッファに 100 個の遷移シーケンスが含まれる

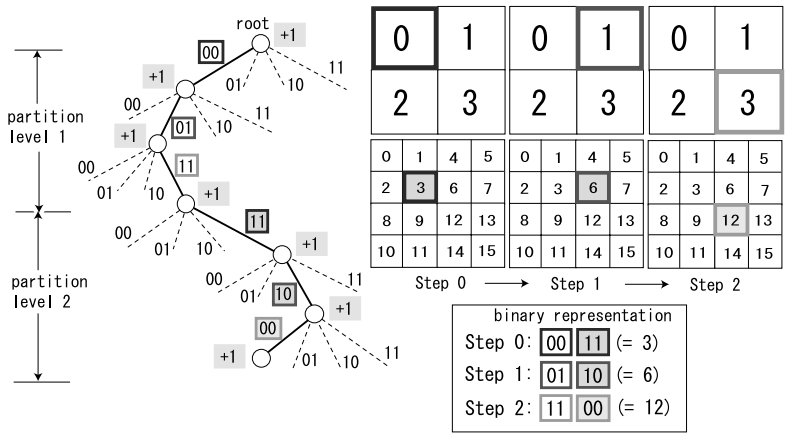

図 4　ヒストグラムの物理構造
Fig. 4　Physical histogram structure.

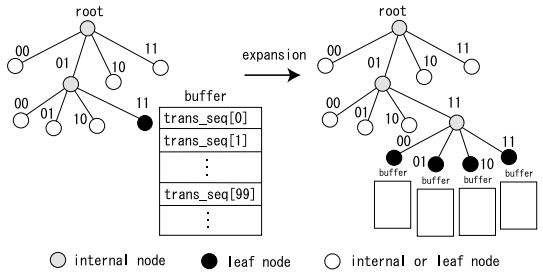

図 5　ノードの展開のためのチェック
Fig. 5　Examination for node expansion.

と想定する．それらのシーケンスを，ステップ 2 の最初の 2 ビットが 00，01，10，11 のどれであるかにより分類し，それぞれのシーケンス数をカウントする．その結果が，それぞれ 25，24，26，24 という値であったとする．このことは，ステップ 0 で領域 $1^{(1)}$ にいて，ステップ 1 で領域 $3^{(1)}$ にいたオブジェクトがステップ 2 で四つの領域（$0^{(1)}$，$1^{(1)}$，$2^{(1)}$，$3^{(1)}$）のそれぞれに進む確率がほぼ等しいことを意味する．一方，カウントがそれぞれ 55，5，40，0 であった場合，カウントの分布には偏りがある．このようなとき，本手法ではこの葉ノードを展開し，新たに四つの子ノードを作成する．そして，バッファ中の遷移シーケンスを対応する子ノードに分配し，それぞれのバッファ領域に格納する．このアプローチにより，移動パターンの分布に応じて木構造が適応的に伸張される．このように木構造を適応的に展開するアプローチは，ストリーム型データに対する決定木構築においても見られる [30]．

木構造の伸張は，遷移シーケンスの追加の結果，木のノード数がユーザから指定された上限 N に達するまで続ける．N に達したら，葉ノードの各バッファ中に含まれる遷移シーケンスの数を四つの分岐ごとにカウントし，それらの値を葉ノード中に記録する．その後，最終的には各バッファ（及びそれに含まれる遷移シーケンス）を削除する．前述の基本構造とは異なり，近似的な木構造では，葉ノードにのみカウントを保持することに注意する．この時点で木の構造は固定され，それ以降に新たな遷移シーケンスが到着した場合，対応する葉ノードのカウント値のインクリメント処理で対応する．到着した遷移シーケンスの総数があらかじめ与えられた定数に達したとき，木構造をファイルに出力し，次のヒストグラムの構築を開始する．

5.1.3　偏りの判定

近似的な木構造の構築においては，カウントの分布の偏りをどのように検出するかがかぎとなり，

- 統計的に明確な基準があること
- 効率的な実装が可能であること

が求められる．後者は，ストリームデータを処理する環境では極めて重要である．このため，本研究では χ^2 適合性検定（χ^2 testing for goodness of fit）[31] を利用する．葉ノードの展開が必要かをチェックする際の，00，01，10，11 のカウントのそれぞれを x_{00}，x_{01}，x_{10}，x_{11} とする．ここでの帰無仮説は，「遷移シーケンスのカウントの分布は一様であり，遷移シーケンスが四つの領域のどれに分類されるかの確率は 1/4 である」である．χ^2 値は次のように計算される．

$$\bar{x} = \frac{x_{00} + x_{01} + x_{10} + x_{11}}{4} \tag{1}$$

$$\chi^2 = \sum_{c\in\{00,01,10,11\}} \frac{(x_c - \bar{x})^2}{\bar{x}} \qquad (2)$$

この値は，自由度 $4-1=3$ の χ^2 分布に従い，有意水準 5%の場合，$\chi^2 > 7.815$ なら分布が均一でないと判定できる．この定数は χ^2 統計表より得られる．

ただし，χ^2 適合性検定は，式 (2) におけるいずれかの x_c の値が小さい場合には利用できないという制限がある．このような状況は，ストリーム的に遷移シーケンスを処理する状況では一般的に発生する．この問題に対応するため，遷移シーケンスの総数が少ない状況では，χ^2 適合性検定ではなく，ノンパラメトリックな検定を行う．この手法は，ロバストなノンパラメトリック検定である 2 項検定（binomial test）[31] の拡張である．詳細は付録において述べる．

5.2 問合せ処理

与えられた遷移シーケンスの出現数をヒストグラムをもとに推定する方法について述べる．問合せシーケンス $3^{(2)} \to 6^{(2)} \to 9^{(2)}$ が与えられたとする．このシーケンスはバイナリー表現では 0011 → 0110 → 1001 となるので，木を

$$\underset{(1,\,0)}{00} \to \underset{(1,\,1)}{01} \to \underset{(1,\,2)}{10} \to \underset{(2,\,0)}{11} \to \underset{(2,\,1)}{10} \to \underset{(2,\,2)}{01}$$

とルートからたどることになる．ただし，00，01，10，11 はたどる枝のラベルを表し，(l, s) は，それが分割レベル l，ステップ s に対応することを意味する．問合せは，状況に応じて以下のように処理する．

（1） 最後の移動 $01_{(2,2)}$ により達したノードが葉ノードであるなら，ノード内のカウント値を返す．

（2） 一方，最後の移動 $01_{(2,2)}$ により達したノードが内部ノードの場合，このノードの子孫を末端までたどり，すべての葉ノードのカウント値を合計して返す．

（3） たどっている途中で葉ノードに達した場合，その問合せシーケンスについては，ヒストグラムは粗い統計情報しかもっていないことを意味する．そこで，近似的な計算を行う．例えば，

$$\underset{(1,\,0)}{00} \to \underset{(1,\,1)}{01} \to \underset{(1,\,2)}{10}$$

とたどった時点で葉ノードとなった場合を考える．このとき，その葉ノードには，その次の四つの分岐 $00_{(2,0)}$，$01_{(2,0)}$，$10_{(2,0)}$，$11_{(2,0)}$ の各部分木に対するカウント値が収められている．そこで，$11_{(2,0)}$ に対するカウント値 C を取り出し，問合せ結果を $C/4^2 = C/16$ と推定する．

この問合せ処理方式を拡張すると，各ステップにおける問合せ領域の粒度が等しくない場合についても結果を返すことができる．例として，問合せ $3^{(2)} \to 1^{(1)} \to 9^{(2)}$ を考える．これは，ステップ 1 に粗い粒度を用いている．この問合せはバイナリー形式で $0011^{(2)} \to 01^{(1)} \to 1001^{(2)}$ と表せるが，これは $0011^{(2)} \to 01{**}^{(2)} \to 1001^{(2)}$ と展開できる．ただし，“**” は 00，01，10，11 の任意のものを表す．この場合，四つの問合せを別々に発行し，それらの和をとることで問合せに答えることができる．

より複雑な問合せも同様のアプローチで処理できる．問合せを基本的な複数の問合せにに分解し，それらの結果の和をとる．後述のように問合せは高速に処理可能であるので，複数の問合せの実行は大きなオーバヘッドとはならないと考える．

5.3 ビットマップの併用

近似的なヒストグラムはコンパクトであるが，誤差を含んでいる．誤差の軽減のため，ビットマップを併用する手法も提案する．ここで考えるビットマップは，データキューブを粗い解像度で実体化したものであり，各セルをバイナリー値で表す．セルの値が 0 である場合，そのセルに対応する遷移シーケンスが全く訪れていないことを意味し，1 である場合は 1 個以上のシーケンスが訪れたことを表す．例えば，$n=2$ 次の連鎖について，分割パラメータが $P=3$ の場合，$R = 2^{2\times 3} = 64$ であるので，ビットマップのサイズは $R^{n+1} = R^3 = 262{,}144$ ビット $= 32$ kByte となる．

問合せ処理では，ビットマップを次のように利用する．問合せの遷移シーケンスが与えられたとき，そのカウント値が 0 であるか否かをビットマップを用いてまず評価する．0 である場合には結果 0 を返して終了するが，そうでなければ，上述の手法によりカウント値を推定して返す．なお，詳細なレベルのビットマップ（例：$P = 4, 5, \ldots$）を用いることでより正確な評価が行えるが，それには余分な格納領域が必要となるため，精度と効率の面でトレードオフが存在する．

6. 実験及び評価

以下では，**5.1.1** で述べた素朴な方式を NAIVE，**5.1.2** で示した近似的手法を APR，**5.3** における近似的な手法とビットマップの組合せを APR-BM と呼び，これらの比較を行う．**6.1** では実験環境について述べ，**6.2** では格納サイズと構築時間について調べる．**6.3**

と **6.4** では，それぞれ問合せ処理時間と精度について評価する．

6.1 実験環境

実験用移動軌跡データの生成のため，Brinkoff による移動オブジェクトのシミュレータ [32] を用いる．このシミュレータは，実際の都市における道路ネットワーク上での多数の移動オブジェクトの移動をシミュレートする．実験データの生成には，ドイツの Oldenburg 市の中心部（約 2.5 km × 2.8 km の領域）の道路ネットワーク（図 6）を利用する．シミュレータのパラメータを調整することで，シミュレーションの各時点において約 1,000 個の移動オブジェクトが常時シミュレーションエリア内に存在するように調整した．シミュレーションデータは 50 単位時間にわたって収集した．1 単位時間は実時間ではおよそ 1 分に対応する．**4.3** で複数の単位時間の粒度を使い分けるアイデアについて述べたが，実験では一つの単位時間のみを使用する．また，移動軌跡データは定常的と考え，各手法に対して単一のヒストグラムのみを構築する．

実験では，シミュレータにより生成された移動軌跡データの各座標値を，最大分割レベル $M = 10$ による領域へとマッピングして用いる．すなわち，最も詳細なレベルでは地図全体が 1,024 × 1,024 の領域に分割され，移動軌跡が表現される．更に，それらの移動軌跡データから，50,000 個の 2 次（$n = 2$）のマルコフ連鎖の遷移シーケンスを抽出して用いる．なお，いくつかの実験では更にその一部を利用する．プログラムは C 言語で作成し，実験は Pentium 4 CPU（3.2 GHz）と 1 GByte のメモリの PC を用い，Windows XP の Cygwin 環境上で実施した．

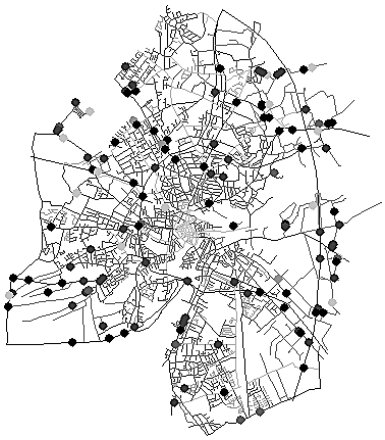

図 6　シミュレーション対象領域
Fig. 6　Simulation area.

6.2 ヒストグラムの構築処理

6.2.1 ヒストグラムのサイズ

表 1 に，三つの手法 NAIVE，APR，APR-BM で構築したヒストグラムのサイズを示す．1 K，10 K，50 K は，入力遷移シーケンスの数を示す(注2)．APR については，割当て可能なノード数の上限 N をそれぞれ 1 K，10 K，50 K と指定した．実際に割り当てられたノード数は，それぞれ 724，6,692，33,844 個であった．すなわち，APR のアルゴリズムは，それぞれの場合において N 個のノードを全部割り当てる必要がないと判断したことになる．APR-BM についても同様のパラメータ設定を行った．ただし，ビットマップ（分割レベル $P = 3$ のバイナリーのデータキューブ：32 kByte）を併用している．

NAIVE のサイズは他と比べ明らかに大きく，コンパクトな統計情報の表現というヒストグラムの目的には合致していないことが分かる．APR-BM はビットマップを併用するが，そのオーバヘッドは小さいことが分かる．以上より，APR 及び APR-BM が，サイズの面で良い性質を有していることが分かる．

6.2.2 ヒストグラムの構築時間

図 7 に，異なるデータサイズにおける NAIVE と APR のヒストグラム構築時間を示す．遷移シーケンス当りの時間であり，それぞれ 10 回の計測の平均値である．APR-BM については，構築時間が APR とほぼ同じであることから，図では省略した．なお，この図では，最大分割レベルを $M = 5$ にした，より粗いレベルの遷移シーケンス集合を入力とした場合の結果も併せて示している．

図 7 より，データサイズが 1 K 及び 10 K の場合には両者の差は顕著ではないが，50 K になると，NAIVE に比べ APR の構築コストが大きくなることが分かる．その理由は APR における偏り判定にある．APR では各葉ノードにおいて，対応する遷移シーケンス集合に

表 1　ヒストグラムのサイズ（MB）
Table 1　Histogram size (MB).

Data Size	NAIVE	APR	APR-BM
1 K	0.35	0.01	0.04
10 K	2.7	0.10	0.13
50 K	9.4	0.52	0.55

(注2)：以降では，kByte の表記を除き $K = 1{,}000$ とする．

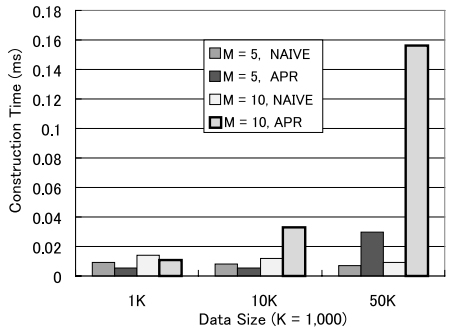

図 7　ヒストグラムの構築時間（遷移シーケンス当り）
Fig. 7　Histogram construction time (per sequence).

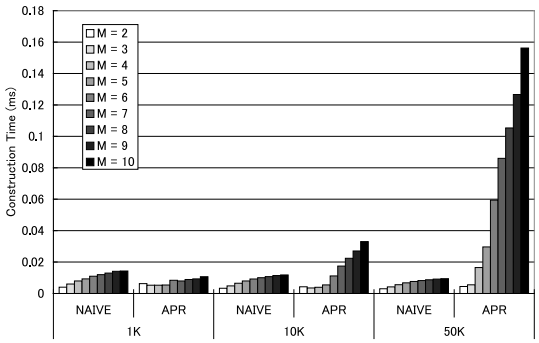

図 8　M を変化させた場合の構築時間
Fig. 8　Construction time for different M values.

偏りがあるかどうかを判定する．付録に示すような効率的な手法を使用してはいるが，判定にはオーバヘッドが存在する．データサイズの増加につれ，対象の葉ノード数も増加することから，構築時間が増加することになる．しかし，図 7 で分かるとおり，APR の遷移シーケンス当りの構築時間は，$M = 10$ の 50 K のデータに対してもわずか 0.16 ms でしかない．交通流のモニタリングのような応用を考えた場合，移動オブジェクトの速度は相対的に遅いため，上記のヒストグラム構築速度で追従が可能であろうと考える．もちろん，大量の数の移動オブジェクトを同時に追跡するような場合にはヒストグラムの構築時間がボトルネックとなる場合も考えられるが，そのような場合にはサンプリングあるいは load shedding [33] などの技術が利用できると考える．

なお，図 7 では遷移シーケンス当りの平均構築時間を示しているが，実際には挿入処理の最初では処理コストは小さく，終りに近づくほど処理コストが増加すると考えられる．そこで，最も処理時間が大きい APR で 50 K のデータを用いた場合について，最後の 10 件の遷移シーケンスを追加する時間を計測した．1 件当りの時間を平均すると，およそ 0.2 ms という結果が得られた．他の設定については，最後の 1 件のデータの追加は 0.1 ms 未満であり，この実験の測定範囲では，データの終りになっても大幅に処理時間が増えないといえる．

図 8 は，入力遷移シーケンスの最大空間分割レベル M を変化させたときの，遷移シーケンス当りの構築時間である．1 K，10 K，50 K はシーケンス数を表す．NAIVE の処理時間が M の増加に対してわずかに増加する程度であるのに対し，APR ではシーケンス数が 10 K，50 K の場合には線形以上のオーダで増加していることが分かる．逆にいえば，M が小さい場合には各手法の違いは相対的に小さいといえる．

6.3　問合せ処理時間

以下では，前節と同じパラメータにより構築したヒストグラムを用いる．1000 個の遷移シーケンスを問合せとして発行し，その処理時間を求める．

（1） シーケンス数 1 K に対するヒストグラムについては，その構築に用いた 1000 個の遷移シーケンスを問合せとする．10 K，50 K のヒストグラムに対しては，構築に用いたシーケンスのうち最初の 1000 個を問合せに用いる．遷移シーケンスの生起頻度に応じて問合せが発生すると仮定すれば，ベンチマークとしては妥当ではないかと考えられる．

（2） レベル 1 から 10 の 10 通りの各問合せレベルについて実験を行う．例えばレベル 2 の場合，入力移動シーケンス $309876^{(10)} \rightarrow 397555^{(10)} \rightarrow 399468^{(10)}$ が与えられたならばレベル 2 の問合せ $4^{(2)} \rightarrow 6^{(2)} \rightarrow 6^{(2)}$ に粗視化して実行する．

図 9 に 1000 件の問合せの処理時間を示す．ただし，処理時間には問合せの前処理・後処理の時間は含まれない．1 K，10 K，50 K は入力遷移シーケンス数である．APR-BM については，コストが APR とほぼ同等であるため省略する．

全体としていえることは，処理時間が非常に高速ということである．1000 件の問合せに対してたかだか 25 ms の処理であることから，1 件当りの処理時間はわずかである．実際の問合せ処理では，問合せプログラムの起動，ヒストグラムの二次記憶からの読込み（主記憶にない場合），結果の出力などの他のオーバヘッドの方が大であると考えられる．問合せ処理について

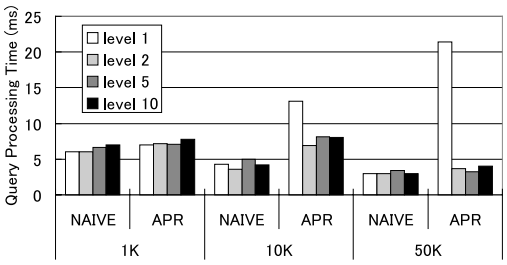

図 9　1000 件の問合せに対する問合せ処理時間
Fig. 9　Query processing time for 1000 queries.

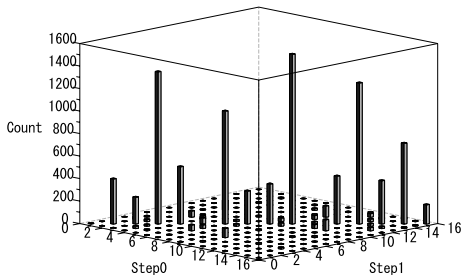

図 10　ヒストグラムのプロット（NAIVE, $n = 1$, $P = 2$）
Fig. 10　Histogram plot. (NAIVE, $n = 1$, $P = 2$)

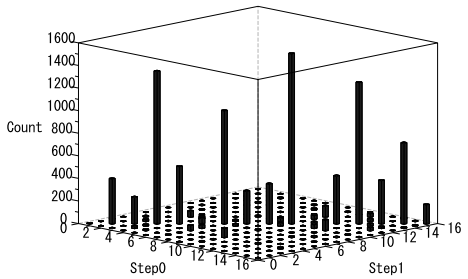

図 11　ヒストグラムのプロット（APR, $n = 1$, $P = 2$）
Fig. 11　Histogram plot. (APR, $n = 1$, $P = 2$)

は，構築処理に比べればリアルタイム性の要求が低いことから，十分な性能であるといえる．

以下では実験結果をより詳細に分析する．図 9 より，NAIVE 方式が APR 方式より一般に優れているといえる．NAIVE 方式の場合，問合せのレベルが深くなるほど，問合せ時間は原理的には増加するはずである．しかし，例えばレベル 1 の問合せとレベル 10 の問合せを比べると，後者の方が 10 倍木をたどる処理を要するが，実際には処理時間にほとんど影響を与えていないことが分かる．これは，木をたどるコストが非常に小さいことを意味している．一方，APR 方式については，部分木の統計情報を再帰的に集約するため，問合せのレベルが浅い方がよりコストが大きいと想像される．実際，図 9 を見ると，10 K，50 K のヒストグラムにおいて，レベル 1 の問合せの処理コストは大きいものとなっている．しかし，レベル 2 以降ではほぼ一定のコストに収束していることが分かる．

また，NAIVE 方式では，1 K，10 K，50 K とヒストグラムのサイズが増えるほど，処理時間が減少傾向にあることが見て取れる．本来は，ヒストグラムのサイズによらずほぼ一定の処理時間と予想されるが，このような傾向がなぜ発生したかについては，明快な理由は得られていない．キャッシュの振舞いなどを含めた解析を今後の課題としたい．レベル 1 の場合を除く APR 方式についても，若干のコストの減少が見られる．APR 方式は NAIVE 方式ほど単純ではないが，NAIVE 方式と共通する理由があるのではないかと想像される．

6.4　ヒストグラムの精度

6.4.1　プロットによる評価

詳細な評価の前に，構築されたヒストグラムの様子をプロットにより観察する．図 10 は，10 K の遷移シーケンスに対し，NAIVE 法を用いて構築されたヒストグラムのプロットである．マルコフ遷移の次数は $n = 1$ 次であり，空間分割レベルは $P = 2$ である．図 11 は同じデータに対する APR によるヒストグラムである．両者の内容がほぼ同等であることが分かる．対角領域のセルが高い値をとる理由は，移動オブジェクトが移動する際に，限られた時間内では同一のセル，若しくは近傍のセルにしか物理的に移動できないことによる．

6.4.2　評 価 尺 度

精度の評価のため，以下の二つの尺度を用いる．*Dist* はユークリッド距離であり，*RelErr* は相対誤差である．

$$Dist = \sqrt{\sum_{i=1}^{R^{n+1}} (ACT_i - EST_i)^2} \tag{3}$$

$$RelErr = \sqrt{\frac{1}{R^{n+1}} \sum_{i=1}^{R^{n+1}} \left(\frac{ACT_i - EST_i}{ACT_i} \right)^2} \tag{4}$$

ここで $R = 2^{2P}$ であり，R^{n+1} は論理的なヒストグラム（データキューブ）のセルの総数を表す．セルには 1 から R^{n+1} の番号が付けられているとする．ACT_i は，セル i の実際のカウント値を表しており，NAIVE 法によるセル値の値である．EST_i は近似されたヒス

トグラムにおけるセル i の推定値である．

式 (4) の計算においては，分母の ACT_i が 0 になることがあるという問題がある．この問題を解決するため，相対誤差の計算では，ACT_i，EST_i の値にラプラス推定（Laplace estimator）による補正を行う．

$$\bar{C}_i = \frac{C_i + 1}{S + B} \times S \tag{5}$$

ただし，C_i はオリジナルの値（ACT_i あるいは EST_i）を表す．B と S は以下のように定義される．

$$B = R^{n+1} \tag{6}$$

$$S = \sum_{i=1}^{B} C_i \tag{7}$$

ラプラス推定についての詳細については [24] を参照頂きたい．

6.4.3 評価結果

典型的なパラメータ設定における評価結果を示す．図 12 は，ユークリッド距離 $Dist$ による APR と APR-BM の比較結果である．50 K 個の遷移シーケンスに対する $n = 2$ 次のヒストグラムを用いている．APR-BM のビットマップは分割レベル $P = 3$ で構築した．式 (3) の計算では，分割レベル $P = 3$ を用いた．すなわち，分割レベル $P = 3$ のすべての遷移シーケンス（例：$1^{(3)} \to 2^{(3)} \to 3^{(3)}$）を列挙し，それらのセル値を推定して距離を計算している．この実験では，同程度の格納サイズでの精度の比較を行うため，最大割り当てノード数 N を調整し，結果として得られるヒストグラムのサイズが 150 kByte，300 kByte，450 kByte，600 kByte にできるだけ近くなるようにした．図より，割り当てるノード数が増えることで精度が改善されるが，その効果は次第に落ちることが分かる．この図では，APR と APR-BM の間の差は顕著ではないといえる．

図 13 は，同じ条件下での相対誤差 $RelErr$ の評価である．APR-BM がよい精度であることは明らかである．この理由は，値 0 をとるキューブセルに対し APR-BM が正確な推定を与えるためである．$Dist$ の場合，大きい値をとるセルが全体のスコアを支配するため，小さい値をもつセルの影響は小さかったが，$RelErr$ の場合には小さい値のセル値の正確な推定がスコアに大きな影響を与える．

二つの評価結果は二つの近似手法の特徴をよく表している．ビットマップの併用は，相対誤差が重要な場合にわずかなサイズのオーバヘッドで性能の向上を達成できる良い手段であるといえる．

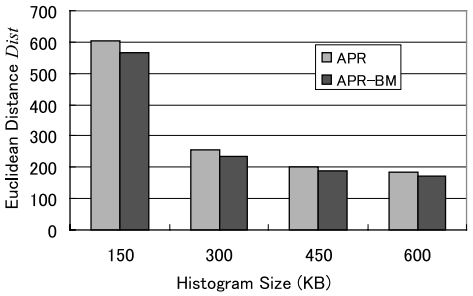

図 12 ユークリッド距離
Fig. 12 Euclidean distance.

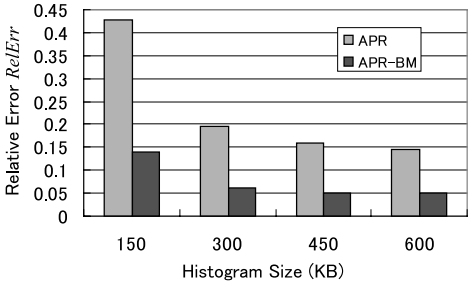

図 13 相対誤差
Fig. 13 Relative error.

7. むすび

本論文では，多数の移動オブジェクトの移動状況をリアルタイムに集約するための，マルコフ連鎖に基づく移動ヒストグラムの構築法について，基本概念，実装方式，及び実験による評価について述べた．移動ヒストグラムは論理的にはデータキューブの構造をもちOLAP 的な分析を可能とし，物理的には木構造により効率的に実現される．物理表現の方式として，素朴な実装方式（NAIVE）に加え，近似を用いたコンパクトな表現（APR），及び近似にビットマップを併用し精度を向上させる手法（APR-BM）を提案した．

近似を用いた APR 及び APR-BM では，ストリーム的に得られる遷移シーケンスに対し，適応的に木を伸張させるアプローチをとった．現在の葉ノードを伸張するか否かを判定するため，χ^2 適合性検定及びそのノンパラメトリック版により統計的な偏りを判定し，カウント値に偏りが見られる場合には木を展開する方式を用いた．これらの手法ではユーザがヒストグラムのノード数の上限値 N を指定することもできるが，N

を指定しない場合でも，カウント値に偏りが見られない場合には展開を行わず，木をコンパクトなサイズにとどめる点に特徴がある．一般のヒストグラム手法では，バケット数（ノード数）の決定はユーザにゆだねられるが，適切にバケット数を指定することは容易ではない．これに対し，統計的な偏りを指標として適切なサイズのヒストグラムを与えることができる点は本手法の特色であり，ユーザの負担の軽減につながる．

実験により，ヒストグラムのサイズに関しては，2種類の近似方式が素朴な方式に比べ顕著に優れていることが示された．ヒストグラム構築に関しては，近似方式の処理時間は素朴な手法よりも大きいが，移動オブジェクトの移動状況のモニタリング要求に耐えられる程度のものであることが分かった．近似手法の精度に関する評価では，近似手法が移動状況の全般的な傾向を大まかにとらえていることをプロットにより確認した．また，ユークリッド距離と相対誤差の2種類の評価法により評価を行い，特に相対誤差においてはビットマップの併用が，わずかなオーバヘッドで大きな効果をもたらすことが分かった．

実験においては，主として $n = 2$ 次のマルコフ遷移に関する結果を示した．提案手法自体は $n \geq 3$ の場合についても原理的には適用可能であるが，次数が増えるとデータ構造のサイズが指数的に増加していくという問題がある．このような問題は，言語処理における N グラム統計量においても同様に見られるものであり [24]，コストに見合うほどの精度の向上が必ずしも得られないことが指摘されている．サイズに制約がある移動ヒストグラムにおいても，高次の遷移データの統計をとるよりも，限られたサイズの中で低次の統計量を正確に表現することの方がより妥当であるといえる．これにより，高次の遷移パターンに対する推定が求められた場合には低次の推定の組合せにより推定を行うことになるが，現実的な解決策であると考えられる．

提案手法では，空間レベルでの粒度を適応的に変化させることにより，限られたサイズの中での精度の向上を目指したが，集積する遷移シーケンスの次数についても適応的に変化させることが考えられる．例えば，どの移動オブジェクトも直線的に移動するような領域については低次の遷移シーケンスを収集し，複雑な移動が見られる領域については高次の遷移シーケンスを収集するなどである．高次の統計量への対応に加え，この問題については今後の課題として検討したい．

その他の研究課題としては，実際の移動オブジェクトの移動データを用いた評価や，処理コストを抑えた上での精度の更なる向上のための方式の改良などが挙げられる．また，本論文では移動ヒストグラムの基本的な機能の提案を行ったが，実際の利用を考えた場合には，利用状況に適したパラメータの設定が求められる．対象となるデータの特性や，利用する側の要求などを考慮したチューニング技術の開発なども今後の研究課題としたい．更に，提案した移動ヒストグラムの手法を用いて，探索的な移動分析を行うためのフレームワークの開発も行いたいと考えている．

謝辞　投稿論文に対し，詳細に査読頂き価値あるコメントを頂いた査読者の方々に感謝致します．本研究の一部は，日本学術振興会科学研究費（15300027, 16500048），文部科学省科学研究費特定領域研究（18049005），栢森情報科学振興財団及びセコム科学技術振興財団の助成による．

文　献

[1] R.H. Güting and M. Schneider, Moving Objects Databases, Morgan Kaufmann, 2005.
[2] C.S. Jensen ed., Special issue: Indexing of moving objects, IEEE Data Engineering Bulletin, vol.25, no.2, 2002.
[3] C.S. Jensen ed., Special issue: Infrastructure for research in spatio-temporal query processing, IEEE Data Engineering Bulletin, vol.26, no.2, 2003.
[4] Y.-J. Choi and C.-W. Chung, “Selectivity estimation for spatio-temporal queries to moving objects,” Proc. ACM SIGMOD, pp.440–451, 2002.
[5] Y.-J. Choi, H.-H. Park, and C.-W. Chung, “Estimating the result size of a query to velocity skewed moving objects,” Inf. Process. Lett., vol.88, no.6, pp.279–285, 2003.
[6] Y. Tao, J. Sun, and D. Papadias, “Selectivity estimation for predictive spatio-temporal queries,” Proc. ICDE, pp.417–428, 2003.
[7] G.J.G. Upton and B. Fingleton, Spatial Data Analysis by Example, Volume II: Categorical and Directional Data, John Wiley & Sons, 1989.
[8] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2005.
[9] Y. Ishikawa, Y. Machida, and H. Kitagawa, “A dynamic mobility histogram construction method based on Markov chains,” Proc. SSDBM, pp.359–368, 2006.
[10] I.F.V. López, R.T. Snodgrass, and B. Moon, “Spatiotemporal aggregate computation: A survey,” IEEE TKDE, vol.17, no.2, pp.271–286, 2005.
[11] S. Acharya, V. Poosala, and S. Ramaswamy, “Selectivity estimation in spatial databases,” Proc. ACM SIGMOD, pp.13–24, 1999.

[12] J. Sun, D. Papadias, Y. Tao, and B. Liu, “Querying about the past, the present, and the future in spatio-temporal databases,” Proc. ICDE, pp.202–213, 2004.
[13] Y. Ishikawa, Y. Tsukamoto, and H. Kitagawa, “Extracting mobility statistics from indexed spatio-temporal datasets,” Proc. 2nd Workshop on Spatio-Temporal Database Management (STDBM’04), pp.9–16, 2004.
[14] Y. Ioannidis, “The history of histograms (abridged),” Proc. VLDB, pp.19–30, 2003.
[15] N. Bruno, L. Gravano, and S. Chaudhuri, “STHoles: A workload aware multidimensional histogram,” Proc. SIGMOD, pp.211–222, 2001.
[16] V. Poosala and Y. Ioannidis, “Selectivity estimation without the attribute value independence assumption,” Proc. VLDB, pp.486–495, 1997.
[17] S. Muthukrishnan, V. Poosala, and T. Suel, “On rectangular partitionings in two dimensions: Algorithms, complexity, and applications,” Proc. ICDT, pp.236–256, 1999.
[18] P. Gibbons, Y. Mattias, and V. Poosala, “Fast incremental maintenance of approximate histograms,” Proc. VLDB, pp.466–475, 1997.
[19] Y. Mattias, J.S. Vitter, and M. Wang, “Dynamic maintenance of wavelet-based histograms,” Proc. VLDB, pp.101–111, 2000.
[20] N. Thaper, S. Guha, P. Indyk, and N. Koudas, “Dynamic multidimensional histograms,” Proc. SIGMOD, pp.428–439, 2002.
[21] J.A. Orenstein, “Spatial query processing in an object-oriented database system,” Proc. SIGMOD, pp.326–336, 1986.
[22] P. Rigaux, M. Scholl, and A. Voisard, Spatial Databases: With Application to GIS, Morgan Kaufmann, 2001.
[23] M. Hadjieleftheriou, G. Kollios, D. Gunopulos, and V.J. Tsotras, “On-line discovery of dense areas in spatio-temporal databases,” Proc. SSTD, vol.2750 of LNCS, pp.306–325, 2003.
[24] C.D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
[25] P. Flajolet, “The ubiquitous digital tree,” Proc. STACS, vol.3884 of LNCS, pp.1–22, 2006.
[26] D.E. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, 2nd ed., Addison-Wesley, 1998.
[27] R. Sedgewick, Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3rd ed., Addison-Wesley, 1998.
[28] H. Samet, The Design and Analysis of Spatial Data Structures, Addison-Wesley, 1989.
[29] J.L. Bentley, “Multidimensional binary search trees used for associative searching,” CACM, vol.18, no.9, pp.509–517, 1975.
[30] P. Domingos and G. Hulten, “Mining high-speed data streams,” Proc. KDD, pp.71–80, 2000.
[31] J.H. Zar, Biostatistical Analysis, 4th ed., Prentice Hall, 1998.
[32] T. Brinkhoff, “A framework for generating network based moving objects,” GeoInfomatica, vol.6, no.2, pp.153–180, 2002.
[33] D.J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, “Aurora: A new model and architecture for data stream management,” VLDB J., vol.12, no.2, pp.120–139, 2003.
[34] 青木繁伸, “正確確率検定（exact test）,” http://aoki2.si.gunma-u.ac.jp/exact/exact.html

付　　録

ノンパラメトリック統計を用いた偏り判定

式 (2) の χ^2 値を用いた偏り判定では，カウント値 $x_{00},\ldots,x_{11}$ の値が小さい場合に正確な判定が行えないという問題がある．本手法の適用状況では，各葉ノードに付随するバッファに，対応する遷移シーケンスが順次到着するため，初期時点では特にカウント値が小さく，偏りの判定が困難となる．そこで，カウント値が小さい範囲にはノンパラメトリック検定を用い，χ^2 値でなく厳密な確率の計算に基づく評価を行う．

ノンパラメトリック検定の一つとして 2 項検定（binomial test）[31] がある．その典型的な利用例として，帰無仮説「コインの表裏が出る確率が 1/2 である」の検定を考える．t 回の試行を行ったとき，表，裏のうち少ない出現回数の方を k とする．例えば $t=10$ としたとき，$k=0$ となる確率は $\Pr(k=0)=2\cdot\binom{10}{0}\cdot(\frac{1}{2})^{10}=0.002$ であり，$k=1$ の場合は $\Pr(k=1)=2\cdot\binom{10}{1}\cdot(\frac{1}{2})^{10}=0.020$ となる．同様に，$\Pr(k=2)=0.088$，$\Pr(k=3)=0.234$ となる．$\sum_{i=0}^{1}\Pr(10,i)=0.021<0.05$ であり $\sum_{i=0}^{2}\Pr(10,i)=0.109>0.05$ であるので，5%棄却域は $[0,1]$ となる．すなわち，$t=10$ 回の試行のうち，少ない方の面の出現回数 k が 0 若しくは 1 のとき帰無仮説を棄却でき，偏りがあると判定できる．

本研究ではこれを多項（multinomial）の場合に拡張する(注3)．x_{00}，x_{01}，x_{10}，x_{11} の四つの面があるサイコロを考える．帰無仮説は「各面の生起確率は 1/4 である」である．例えば，$t=8$ 回の試行の後，各面の生起数が $x_{00}=1$，$x_{01}=6$，$x_{10}=0$，$x_{11}=1$ であったとする．これらのカウントを大きい順に並べた $(6,1,1,0)$ を生起パターンと呼ぶ．帰

(注3)：関連するアイデアが [34] で簡単に触れられている．

無仮説が正しければ，このパターンの生起確率は $\Pr(6,1,1,0) = 4 \cdot 3 \cdot \frac{8!}{6!1!1!0!}(\frac{1}{4})^8 = 0.0103$ である．このパターンより出現がまれなパターンは，$(6,2,0,0)$，$(7,1,0,0)$，$(8,0,0,0)$ であり，それらの確率はそれぞれ $\Pr(6,2,0,0) = 0.00513$，$\Pr(7,1,0,0) = 0.00146$，$\Pr(8,0,0,0) = 0.0000610$ である．これらの確率の和は $\Pr(6,1,1,0) + \Pr(6,2,0,0) + \Pr(7,1,0,0) + \Pr(8,0,0,0) = 0.0169 < 0.05$ であるので，パターン $(6,1,1,0)$ は 5%の棄却域に含まれるまれな生起パターンであるといえる．

$t = 1, 2, \ldots$ に対する生起パターンについて同様の分析を行うと，5%の棄却域で棄却可能なパターンを以下のように導ける．

- $t = 1, 2, 3$：該当なし
- $t = 4$：$(4,0,0,0)$
- $t = 5$：$(5,0,0,0)$
- $t = 6$：$(6,0,0,0)$, $(5,1,0,0)$
- $t = 7$：$(7,0,0,0)$, $(6,1,0,0)$, $(5,2,0,0)$
- ...

実装においては，$t = 52$ までこのような導出を行った．それ以降の大きい t の値については，χ^2 適合性検定が安全に利用可能である(注4)．

ヒストグラム構築においては，棄却域に含まれるパターンをあらかじめ求めて表に格納しておくことで，偏りの判定処理の効率化を図る．加えて，遷移シーケンスが到着するたびに判定を行うという本手法の適用状況を考慮することで，この表のサイズを削減できる．例えば，$t = 5$ については，上記の棄却パターン $(5,0,0,0)$ を表に収めておく必要はない．なぜなら，$(5,0,0,0)$ というカウント数の分布に達するには，$t = 4$ の時点で $(4,0,0,0)$ という分布でなければならない．しかし，$(4,0,0,0)$ は棄却対象のパターンであるため，Prentice Hall, $(4,0,0,0)$ という分布が得られた時点でノードの展開が行われる．よって，$(5,0,0,0)$ というパターンは実際には発生し得ない．このような点を考慮して最適化を行った結果，表に収める棄却パターンを 465 個に削減することができた．

(注4)：詳細については省略するが，$t = 52$ に至るまでに稀なパターンが排除されるため，$t \geq 52$ については四つのカウント値のいずれも 5 以上であることが保障でき，χ^2 検定の適用範囲になる．

（平成 18 年 5 月 15 日受付，8 月 17 日再受付）

石川　佳治　（正員）

1989 筑波大・第三学群・情報学類卒．1994 同大大学院博士課程工学研究科単位取得退学．同年奈良先端科学技術大学院大学助手．1999 筑波大学電子・情報工学系講師．2004 同助教授．2006 名古屋大学情報連携基盤センター教授．博士（工学）（筑波大学）．データベース，データ工学，情報検索等に興味をもつ．日本データベース学会，情報処理学会，人工知能学会，ACM，IEEE CS 各会員．

町田　陽二

2004 中大・理工・経営システム卒．2006 筑波大大学院理工学研究科修士課程了．現在，（株）日立情報システムズ事業開発本部にて勤務．時空間データベースに興味をもつ．日本データベース学会会員．

北川　博之　（正員：フェロー）

1978 東大・理・物理卒．1980 同大大学院理学系研究科修士課程了．日本電気（株）勤務の後，1988 筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学大学院システム情報工学研究科教授，及び同大学計算科学研究センター教授（併任）．理博（東京大学）．異種情報源統合，文書データベース，WWW の高度利用等に興味をもつ．著書「データベースシステム」（昭晃堂），「The Unnormalized Relational Data Model」（共著，Springer-Verlag）など．日本データベース学会副会長，情報処理学会フェロー，ACM，IEEE-CS，日本ソフトウェア科学会各会員．