空間統計処理のための Geohash を用いたデータベースシステムの提案と評価

小川 輝樹† 伊藤 嘉博† 中村 嘉隆†† 高橋 修†† 白石 陽††

近年，スマートフォンなど多くのセンサを搭載したデバイスを持たせた自動車や自転車，人間などの移動体でセンシングを行う技術としてモバイルセンシングが注目されている．これらの移動体はモバイルセンサと呼ばれ，モバイルセンサからセンサデータを収集し，格納することで固定センサと補完的に利用することが期待されている．その応用として，モバイルセンサ，即ちユーザの一日の行動履歴や動作パターンをもとにした統計処理や統計処理に多次元の要素を追加する空間統計などがある．しかし，統計処理や空間統計において，空間統計を把握する際の処理時間が増大するという問題がある．本稿では，問い合わせ処理速度向上のために，階層的な空間データ構造である geohash を用いたデータベースシステム及び空間統計データ生成・管理手法を提案する．また，提案システムを実装し，その提案システムの有用性について検討する．

Proposal and Evaluation of a database system for processing using spatial statistics Geohash

TERUKI OGAWA† YOSHIHIRO ITO† YOSHITAKA NAKAMURA†† OSAMU TAKAHASHI†† YOH SHIRAISHI††

Recently, mobile sensing has been attracting attention as a technique for performing sensing at car and bicycle, person such as mobile object using smartphones is mounted with many sensors. These mobile objects are called mobile sensor. Collecting and storing sensor data from mobile sensor is expected complementary to use with fixed sensor. As an application, There, other words user is such as service of statistical processing and spatial statistics using statistical processing based on action history of a day and behavior patterns. However, there is a problem of query processing time increases due to be in spatial statistics, sensor data will be stored in a huge amount. In this paper, in order to improve the speed of query processing, we propose a method for data generation and management and improve the processing speed of the database statistics system using geohash is hierarchical spatial data structure. In addition, we designed the system, to examine the usefulness of the proposed system.

†公立はこだて未来大学大学院 システム情報科学研究科
Graduate School of Systems Information Science, Future University Hakodate
††公立はこだて未来大学大学 システム情報科学部
School of Systems Information Science, Future University Hakodate

1. はじめに

近年，高機能化に伴い多くの機能を始めとし，各種センサ等を備えた次世代移動端末機であるスマートフォンが急速に普及している．さらに，スマートフォンなど多くのセンサを搭載したデバイスを用いた自動車や自転車，人間などの移動体でセンシングを行う技術として参加型センシングの Participatory Sensing[1]や People Centric Sensing (PCS)[2]などがある（以下，モバイルセンシングと呼ぶ）．モバイルセンシングによって得られたセンサデータや位置情報を用いた研究やサービスとして，環境情報収集[3]やユーザの行動履歴収集[4]，リモート見守りシステム[5]などが挙げられ，様々な分野で注目されている． Sun SPOT（Sun Small Programmable Object Technology）[6]のように特定の場所に固定して設置した上で環境情報を取得する固定センサとは異なり，モバイルセンサはセンサ自身が移動できるため，センサデバイスを設置できない場所でも細粒度の環境情報を得ることが可能となる．そのため，こうしたモバイルセンサからセンサデータを収集し，データベースに格納することで従来の固定センサのセンサデータと補完的に利用することが期待されている．

収集したデータの活用として，GPS やスマートフォンなどを保持したユーザの一日の行動履歴や動作パターンをもとにした統計データや統計データに多次元の要素を追加した空間統計などがある．空間統計とは，統計処理に多次元の要素を組み合わせた統計処理であり，モバイルセンサで観測された位置情報に基づいて計算した人口密度や滞留人口などが挙げられ，近年では基地局から携帯電話の位置情報の統計を取得するモバイル空間統計[7]などが研究として注目されている．空間統計は，個体や集合体に分類しデータ分析することができ，さらに，時系列などの要素を組み合わせることでより多くのデータ分析が可能となる．

近年，このような空間上に分布したデータをデータベース上から効率的に検索する提案が行われている．しかし，空間統計を把握する場合，様々な範囲を指定して統計を取得する必要がある．その点において従来手法では空間を分割するために様々な処理を行う必要があり，その結果処理時間が増大するといった問題がある．

そこで，本稿では問い合わせ処理速度向上のために，階層的な空間データ構造である geohash を用いたデータベースシステムを提案する．提案システムは，処理速度の向上及び空間統計データ生成・管理を目的とし，有用性について検討する．

2. 関連研究

本章では，2.1 節でモバイルセンシングについて，2.2 節と 2.3 節で空間インデックスに関連する研究について述べる．空間インデックスとは，地図のような 2 次元，

3 次元のデータを扱う空間データベースの技術で，データを高速に検索するために使われる．

2.1 モバイルセンシング

モバイルセンシングとは，第 1 章で述べたようにモバイルセンサを利用したセンシング技術であり，例えば，センサを持った車やスマートフォン，人，自転車もモバイルセンサとして挙げられる．モバイルセンサは環境の出来事や特性を人間などが自ら操作しデータを送る参加型センシング（Participatory Sensing[1]）や People Centric Sensing (PCS)[2]が注目されている．PCS は，人間などが意識せず自動的に環境情報を送るものであり，監視したいエリアにあらかじめセンサを設置することなく，そこを通りがかったモバイルセンサ（人や自動車）が持つセンサにより環境情報をセンシングしデータを送る．この技術を活用することで，広大な領域の環境情報を低コストで収集することできる[8].

このように，モバイルセンサが取得したデータは幅広く活用されており，さらに時系列などの要素を追加した技術として空間統計が注目されている．空間統計としては，センサの分布図や人口密度，滞留人口などが挙げられ，その結果としてデータを客観的に観察し，分析することができる．モバイル空間統計[7]のように，細かい時間単位での人口分布や滞留人口などを把握できることは，社会にとっても有用であるため，本稿ではモバイルセンシングを利用した空間統計処理を対象とする．

2.2 I-Tree

文献[9]では，空間時系列データの高速な検索のためのデータ構造として I-Tree を提案している．I-Tree では，様々なセンサの空間時系列データを効率的に処理することにより，空間領域において設置されたセンサの情報を容易に所得し分析することを目的としている．I-Tree は，空間時系列データを階層的な細分化が可能な文字列マップ表現に変換し，木構造を用いて管理するものである．I-Tree が対象とするデータは，状態が時間とともに変化する空間データであるが，空間自体の位置や形状は変化しないものとしている．

I-Tree は，図 1 のように矩形を等分割し，等分割した領域についてそれぞれ時系列データをまとめて取り扱う．その後，対応する領域の要素として 2 進数で番号を振り分け用いる．振り分けた 2 進数を 10 進数に変換し，各要素を結合して索引文字列 iStr を作成する．

検索に関しては，検引の葉ノードに格納された iStr を利用してデータベース中に格納された実際の空間時系列データにアクセスする．

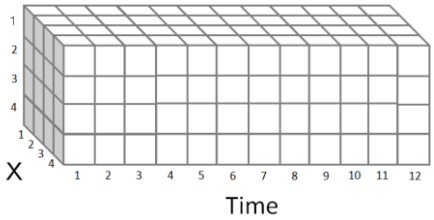

図 1　I-tree における空間時系列データのイメージ

しかし，I-Tree では，空間の矩形が可変的でないため，空間統計情報を柔軟に取得することができない．さらに，索引はあらかじめ用意されているデータから作成しているので新たなデータが登録されると索引を作成し直す必要があり，再構築に大幅な時間がかかる．以上のことから，可変的空間における空間統計情報の取得では，I-tree を用いた検索は適していないと言える．

2.3 Geohash

geohash とは，位置情報をハッシュ値で表す階層的な空間データ構造である．緯度経度データをハッシュ値である文字列（以下，geohash 文字列と呼ぶ）に変換し，地図などの空間情報データをメッシュ状に分割する技術であり，様々な研究や応用システムで使われている[10][11]．文字の地図表示を図 2 に示す．図 2 は，文字をメッシュ表示した図で，(b)は(a)の文字列に 1 文字追加した図となっている．このように，geohash 文字列の長さを変化させることで任意の大きさのメッシュが表現可能となる．geohash は 32 文字の英数字からなり，奇数の場合は縦長，偶数の場合は横長になる．それぞれの文字が対応する 32 (4×8) 分割メッシュを図 2 の(a)に示す．

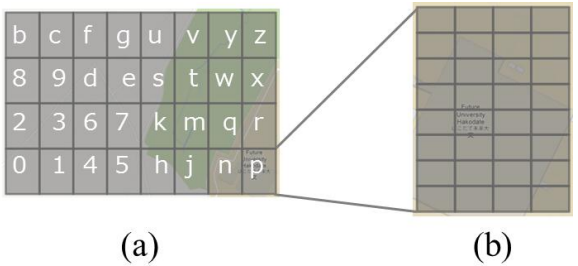

図 2　メッシュ表示（ (a)縮小表現 (b) 拡大表現）

geohash の最大の特徴は，文字の長さによって精度が変わることである．geohash は元々点データを扱ったハッシュ関数であるため変換した文字長を増やすことによって軌跡データを細粒度まで表示可能となる．さらに，geohash は領域をグリッド状に表示できるため文字の長さを変えることで位置をあいまいにできるといった特徴もある．Martins らは geohash を用いて XSLT/XQuery エンジンに地理情報を処理する機能を作成している[14].

Geohash はデータベースから，“xn76ggrw”の情報が知りたい時「xn76gg」といった指定された geohash 文字列（以下，geo-string と呼ぶ）で前方一致検索することで，メッシュ内のデータを簡単に抽出ことができる[12].“xn76gg”のメッシュ表現を図 3 に示す.

図 3　“xn76gg“のメッシュ表示

3. 目的と課題

空間インデックスである I-Tree では，あらかじめ指定された範囲内での空間に分布したセンサデータの検索は可能だが，矩形範囲が可変的に設定できないため，空間統計での検索には適さない．一方，geohash は空間統計データの矩形範囲を柔軟に指定できるため，適した空間インデックスと言える．これらのことから，本稿では従来の手法では空間統計を把握する際の処理時間が増大すること，モバイルセンシングを利用した空間統計データを作成することを課題とし，それぞれの課題を解決する上での問題を考察する．今回は集団に対する統計処理を行い，さらに時系列を組み合わせることとした.

空間統計を把握する際の処理時間増大の課題には，まず緯度経度を geohash により変換した値 geohash 文字列を使用する．geohash は，位置情報を文字列で扱う処理となるためデータベースシステムでは前方一致検索が可能となる．これにより，検索処理の向上が期待できる．さらに，空間統計データの生成では，geohash を用いることで検索範囲内のモバイルセンサを取得し，32 分割された範囲内の人口数のカウントを行う．その結果をアプリケーションに返す.

以上のことから，本稿では，geohash を用いたデータベースシステムの処理速度向上及び空間統計データ生成・管理手法を提案し構築することで，蓄積されたセンサデータが膨大な量になることが原因で問い合わせの処理時間が増大する課題を解消できると考えた．また，このデータベースシステムはユーザ（アプリケーション）から点により位置を指定しデータ検索リクエストがあった場合，以下の 3 つの処理を実行する必要がある.

(1)　指定の点の緯度経度を geohash により変換した geo-string でデータベースにアクセスする処理
(2)　データベースで取得したデータから統計データを作成する処理
(3)　時系列により時間の範囲を指定し空間統計データを作成する処理

(1)の処理では，geo-string でデータベースにアクセスを行う．そのため，メッシュ内のデータを検索するのは前方一致検索を行い，データを取得する必要がある．また，(2)の処理では，データベースから取得データを属性別に分けて出力する必要がある．さらに，そこから統計データを作成する処理が必要となる．さらに，(3)の処理では，年・日付・時刻の指定し，新たにデータを作成する必要がある．これらの処理を含んだデータベースシステムを設計・実装することで，センサデータを利用した空間統計データ生成・管理が実現できると考えられる.

4. 提案方式

本章では，第 3 章で述べた問題を解決し，センサデータを利用した統計データを作成するためのデータベースシステムの設計と詳細について述べる.

4.1 提案システムの構成

本提案システムは，モバイルセンサとモバイルセンシングデータベース（以下，MSDB と呼ぶ），問い合わせ処理システムの 3 つの要素からなる．モバイルセンサは，位置情報・センサデータなどをデータベースに提供する．MSDB は，位置情報・時刻・モバイル ID・センサデータを管理する．問い合わせ処理システムは，MSDB から習得したデータを処理する.

図 4 に，提案システムの全体構成を示す．緯度経度とその緯度経度を geohash により変換した文字列（以下，geohash 文字列と呼ぶ）を同時に登録する．MSDB に登録されたデータを検索する際，検索対象を geohash 文字列にすることで検索処理の向上を図る．クエリを送る際，地図上で範囲や点を指定し，その指定した範囲や点の緯度経度を Geohash を用いて文字列に変換し，その文字列を用いて MSDB に問い合わせを行う．Geohash を用いて変換された文字列を，以下，geo-string と呼ぶ．

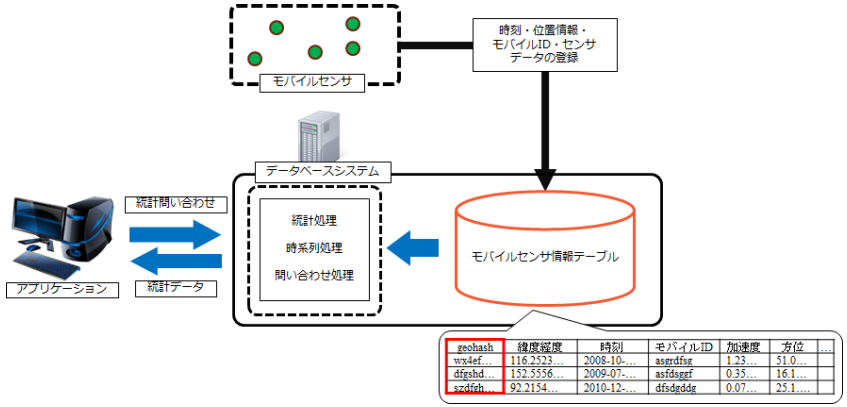

図 4　提案システム全体の構成図

本提案システムは，問い合わせ処理速度の低下を抑制するために geohash を用いる．緯度経度を geo-string にすることにより，文字列の長さを増やすことができるため，指定範囲のデータを容易に絞り込むことができる．そのため，MSDB への問い合わせ処理が従来の緯度経度をもととした検索システムより向上する．さらに，空間統計データを地図に表示するとき，地図に各メッシュに分割して表示するため検索結果のモバイルセンサをメッシュ毎に振り分ける．モバイル ID が同じモバイルセンサを一つにまとめることで絞り込んだメッシュ内の人口数の統計データを取得することができる．

4.2 データベース構造

提案システムの MSDB のレコードは，経度緯度と geohash 文字列，センサデータ，モバイル ID，時刻の 5 つから構成される．

まず提案システム内の MSDB にモバイルセンシングの各センサデータをテーブルへ格納する．

さらに，緯度経度のデータから geohash 文字列に変換する．その変換した文字列の情報を登録したテーブルを格納する．モバイルセンサ情報テーブルの定義の詳細を表 1 に示す．

表 1　モバイルセンサ情報テーブル

属性名	説明
Geohash 文字列	緯度経度を geohash により変換した文字列
緯度経度	モバイルセンサの位置情報（点データ）
計測時刻	センサデータを観測した年・日付・時刻
モバイル ID	モバイルセンサの識別番号
センサデータ	各センサが取得したデータ

Geohash 文字列は，登録されている緯度経度から geohash により文字列に変換した値である．緯度経度は，モバイルセンサの位置情報を点データである．センサデータは，モバイルセンサから取得可能な加速度，方位等といったセンサデータである．センサデータは複数のデータが登録可能であるが，現段階では加速度センサと方位センサの 2 つを登録している．計測時刻はユーザがセンサデータを取得した日付と時刻データである．表 2 にモバイルセンサ情報テーブルの具体的な例を示す．

表 2　モバイルセンサ情報テーブルのレコードの具体例

Geohash 文字列	緯度経度	時刻	モバイル ID	加速度	方位	…
wx4ef…	116.2545…	2008-10-…	asgrdfsg	1.23…	51.0…	
dfgshd…	152.5352…	2009-07-…	asfdsggf	0.35…	16.1…	
szdfgh…	92.25215…	2010-12-…	dfsdgddg	0.07…	25.1….	

4.3 範囲検索

本節では，問い合わせ時の検索 4.3.1 項は文字列による範囲指定，4.3.2 項はポイントとレベルによる範囲指定について述べる．

問い合わせのパラメータは geo-string，時間，モバイル ID であり，geo-string は S，時間は TI，モバイル ID は mID とする．さらに，時間は，開始時刻（Ts）と終了時刻（Te）によって指定する．以下に関係式を示す．

$$Q =< S, TI, mID > \quad (1)$$

$$TI = [Ts, Te] \quad (2)$$

4.3.1 文字列による範囲指定

本項では，文字列指定による範囲検索について述べる．文字列指定では，アプリケーションで指定した範囲を文字列変換した geo-string（*s*）をもとにして，MSDB に対して前方一致検索で問い合わせを行う．前方一致検索とは，先頭文字から数文字をあらかじめ固定して，その部分文字列が一致するかどうか調べることによって検索する処理なので，検索パラメータの文字列以降の文字が何であろうと関係なく対応するメッシュに含まれるすべてのデータを検索することができる．したがって，範囲検索を容易に行うことができ，データの絞込みが可能となる．図 5 に文字列指定による検索の流れを示す．

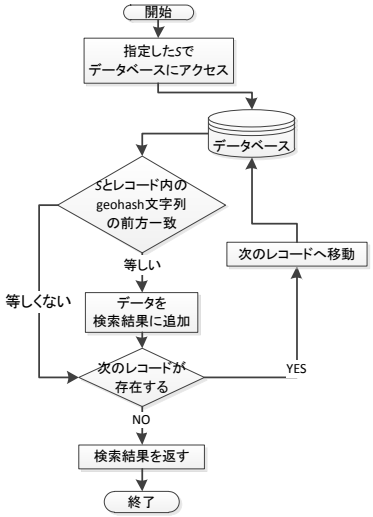

図 5　文字列指定による検索の流れ

(1) アプリケーションで，文字列 *s* を指定する
(2) *s* で MSDB に問い合わせをし，最初のレコードを調べる
(3) *s* とレコード内の geohash 文字列を前方一致で比較する
(4) 前方一致すれば検索結果にそのレコードのデータを検索結果に追加する
(5) 次のレコードが存在すれば(3)の処理を行う
(6) 存在しなければ検索結果を返し終了する

4.3.2 ポイントとレベルによる範囲指定

本項では，ポイントとレベルによる範囲検索について述べる．ポイントは，アプリケーションで指定した地図上の位置のことである．レベルは，geohash の階層を表し，文字列の先頭からの文字数を決めることでレベルを指定することができる．このレベルを指定する文字数を geo-len と呼ぶ指定した位置の緯度経度を *s* に変換しさらに図 6 のように geo-len によりレベルを指定する．geo-len によって *S* の先頭の部分文字列を指定することができる．その部分文字列を *PS* とする．こうすることで，geohash の階層構造を利用し，各レベルでの 32 分割されたメッシュ内に存在するモバイルセンサのデータを取得することができる．図 6 に文字数（geo-len）による階層のレベル指定のイメージを示す．実際には 32 分割のメッシュであるが，図 6 では簡単化のため各メッシュが 4 分割されるものとしている．図 7 にポイントとレベル指定による検索の流れを示す．

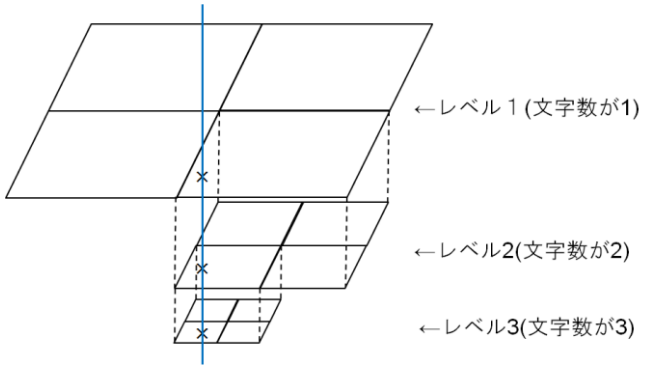

図 6　g 文字数（geo-len）による階層のレベル指定のイメージ

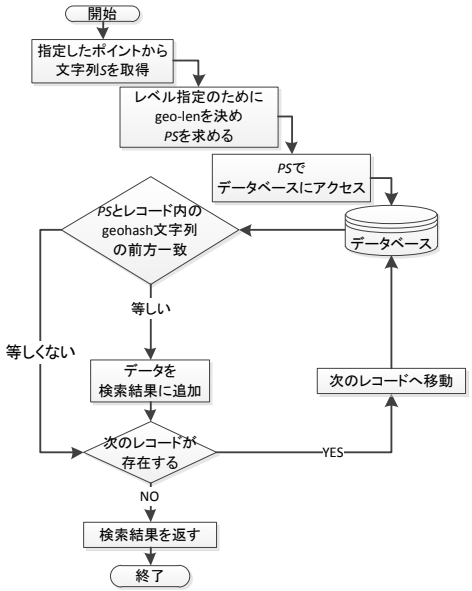

図 7　ポイントとレベル指定による検索の流れ

(1) アプリケーションで指定したポイントを文字列 *S* に変換する
(2) レベル指定のために geo-len の数字を決め，*PS* の文字数を決める
(3) *PS* で MSDB は問い合わせをし，最初のレコードを調べる
(4) *PS* とレコード内の geohash 文字列を前方一致で比較する
(5) 前方一致すれば検索結果にそのレコードのデータを検索結果に追加する
(6) 次のレコードが存在すれば(4)の処理を行う
(7) 存在しなければ検索結果を返し終了する

4.4 空間統計処理

本節では，空間統計処理の詳細について述べる．提案システムでは，空間統計処理の例として，geohash 文字列により範囲を指定し，その範囲内のモバイルセンサの数の分布を示す場合を考える．4.3 節で説明した範囲検索により取得したデータを 32 分割したメッシュ毎にモバイルセンサを分ける必要がある．そこで，*PS* に一文字追加することで，その範囲を 32 分割した各メッシュのいずれかを表現できる．*S* による前方一致検索，一文字追加した文字列による 32 個すべてのメッシュに対して，図 8 の処理を行うことで，そのメッシュに存在する *mID* の数を計算することできる．検索パラメータに指定した範囲内のすべてのメッシュに対して数を計算することで，メッシュ状の分布図を作成できる．

まず，前方一致検索によって検索されたレコードに含まれる *mID* が，そのメッシュの *mID* リストに登録されているかチェックする．*mID* リストとは，メッシュ内に存在する *mID* を管理するリストであり，検索時に毎回更新される．

mID リストは，メッシュごとに作成され，初期値は空である．*mID* リストに前方一致検索によって得られたレコードの *mID* が存在しなければ登録し，逆にすでに存在した場合は登録しない．この処理を行うことで，*mID* の重複を避け，指定した範囲内のメッシュ状のモバイルセンサの数の分布が取得可能となる．モバイルセンサが人に対応していれば，この空間統計処理の結果は，範囲内の人口分布を示す．図 8 に空間統計処理の流れを示す．

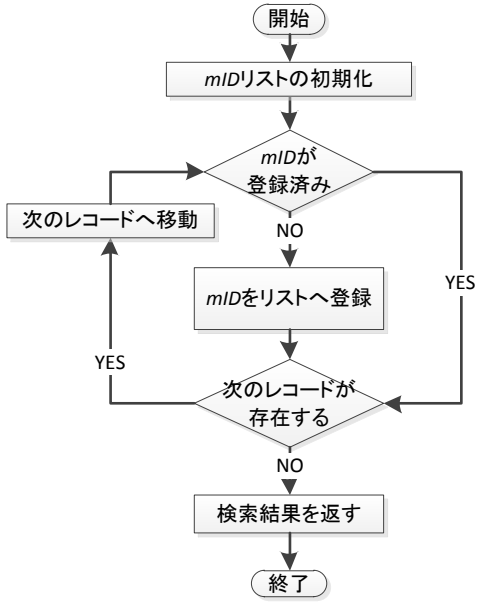

図 8　空間統計処理の流れ

(1) *mID* リストにレコードの *mID* が登録されていなければ登録する
(2) 既に登録済みなら(3)に移る

(3) 次のレコード行が存在するなら(1)へ移る
(4) 存在しなければアプリケーションへ統計結果を返し，終了する

4.5 時系列処理

本節では，時間 *TI* を検索パラメータとして指定時の検索処理について述べる．4.4 節の統計処理に加えて，より詳細なデータの取得を目的とするのが時系列処理である．単に統計処理を行い結果のデータを表示するだけでは最初に登録された古いデータの表示になってしまう．そこで，この問題の解決のために時系列処理を取り入れることでより細かいデータの取得が可能になり，さらにアプリケーションで開始時刻 *Ts* と終了時刻 *Te* を指定することで詳細なデータの取得可能となる．図 9 に時系列処理の流れを示す．

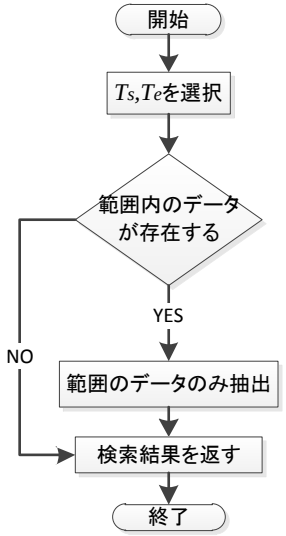

図 9　時系列処理の流れ

(1) アプリケーションが *Ts*,*Te* を指定し，情報を問い合わせシステムへ送り MSDB へ問い合わせを行う．
(2) 取得したデータから範囲内のデータがあれば(3)へ，なければエラーを返し終了する
(3) 指定範囲内のデータのみ抽出し，検索結果を書き込んでいく，すべての処理が完了したらアプリケーションへ検索結果を返し，終了する．

5. 実装および評価

4 章で述べた提案システムの有効性を検証するにあたり，実際に提案方式を用いることで統計データを算出し，その結果について評価を行う必要がある．そこで，提案システムの機能及び処理を実装したデータベースシステムのプロトタイプシステムと算出した統計データを表示する評価アプリケーションの構築を行った．

5.1 実装環境

本節では，実装環境の詳細について述べる．まず，実験に用いるマシン性能とデータベースシステム，評価システムのバージョンを表 3 にまとめて示す．

表 3　実装環境

ハードウェア	
OS	Windows 7 Professional 64bit
CPU	Intel Core i7
RAM	8.00GB
HDD	500GB
データベースシステム	
DBMS	PostgreSQL 8.4
プログラム	Java Servlet
評価用システム（表示部分）	
プログラム	Java Server Pages（Java Script）
ブラウザ	Mozilla Firefox 9.0.1

提案システムでは位置情報である緯度経度から geometory 型に変換し，さらにそこから geohash 文字列に変換する必要があるため，DMBS には地理情報の計算を SQL で行う PostGIS を標準で利用できる PostgreSQL を採用した．また，各機能は Java Servlet で記述した．生成した統計データの表示を行う評価アプリケーションは PC の上の Web ブラウザで表示を行うため，Java Server Pages（JSP）で記述した．Web ブラウザ上の地図表示には Google Map API[13]を用いた．

5.2 統計データ生成に用いたデータ

統計データ生成に用いたサンプルデータは，GeoLife プロジェクトで収集された GeoLife GPS Trajectories[14]のサンプルデータを用いた．GeoLife GPS Trajectories は 2007 年 4 月から 2011 年 10 月の 167 人，計 17,355 本の軌跡データを収集したデータである．Geohash 文字列は，緯度経度から文字列を生成するためには緯度経度を geometry 型に変換し，それを geohash 文字列に変換するという処理手順を踏む必要がある．そのため，まず緯度経度のみを POINT データとして格納する．その後，geohash 文字列を新たに作成し格納し直す．この軌跡データに個別にモバイル ID を振り分け作成した CSV ファイルのデータを表 4 に示す．

表 4　CSV ファイルのデータ

geohash	20 桁の geohash 文字列を格納
緯度経度(geometry 型)	緯度経度を geometry 型に変換した値を格納
モバイル ID	1~順に数字を割り振り格納
Timestamp	年・日付・時刻を格納

表 5 の CSV ファイルのデータは，1 列目が geohash 文字列となっており，次に緯度経度を PostGIS の geometry 型である POINT に変換した値，モバイル ID，時刻，センサデータとなっている．このデータを MSDB に格納しておき，アプリケーションからの問い合わせ処理時に利用した．

5.3 プロトタイプシステムの構築

MSDB のテーブルは 4.2 節で述べた表 1 のテーブルである．統計データ表示のアプリケーションへの提供には，どのようなアプリケーションでも利用しやすいように，記述の汎用性の高いマークアップ言語である XML を用いた．問い合わせ処理システムにより作成される XML の内容を表 5 に示す．XML の構成としては，親要素である features タグの中に point 1 つのデータを記述した point タグを子要素として記述している．point タグも子要素である mobileid タグ，geohash タグ，latlng タグを持っており，そこにはモバイル ID，geohash，緯度経度のデータが記述されている．

実装した評価用システムでは，Java Script 上で getElementsByTagName メソッドを用いることで XML に記述されたポイントの緯度経度を取得し，それらの位置情報を Google Map API に渡すことで，Google Map 上にメッシュ分割した geohash とポイントとして地図上にオーバーレイ表示している．

表 5　XML の内容

タグ	内容
<features>	属性<point>が入る
<point>	属性<mobileid>,<geohash>,<latlng>,<timestamp>が入る
<mobileid>	モバイルセンサの ID が入る
<geohash>	Geohash 文字列が入る
<latlng>	Geometry 型を緯度経度に戻した値が入る
<timestamp>	年・日付・時刻が入る

5.4 評価アプリケーションの構築

評価アプリケーションは JSP を用い，Google Map 上にメッシュごとの統計データを表示するシステムを構築した．統計処理を行ったデータをアプリケーションに送りメッシュ内のモバイルセンサ数に応じて色分けをして統計データを地図上に表示する．赤枠の指定範囲の geohash を”wx4ewg”と geo-string で表現できる．この geo-string を前方一致検索で MSDB に問い合わせ, 統計処理を行った統計データを図 10 に示す．表示されるマーカの一つ一つがモバイルセンサとなる．

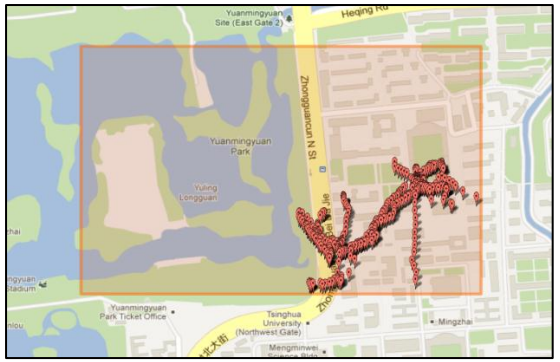

図 10　実行画面

さらに，時系列処理の部分について説明する．図 11 に示すように，選択したポイントの表示させるメッシュ領域の矩形の大きさの選択と，データの絞込みを行う際に年・日付・時刻をそれぞれ指定することでより詳細にデータを取得できる．

図 11　時系列の選択部分

5.5 評価

実際に統計データを生成し，評価を行う．統計データ生成に用いるデータは，2008 年 4 月~2010 年 11 月の約 2 年 6 ヶ月間の 280 万(約 603MB)の位置情報データを採用した．

評価の内容としては，まずアプリケーションからの問い合わせに対して検索・XML への書き込み処理までかかる時間を計測する．次に，時系列処理を加えた処理として正確に取得できているか検証する．

はじめに，アプリケーションからの問い合わせ処理で前方一致検索の処理時間を計測した．具体的には，文字列長を 1 文字～5 文字まで値を変化させ，それぞれの文字列に対して 10 回の試行を行った．geo-string である“w”，“wx”，“wx4”，“wx4e”，“wx4ew”についての平均検索処理時間，geo-string と同じ大きさの範囲の緯度経度をもととした平均検索処理時間を図 12 に示す. 図の s は 1 秒である second を表している．

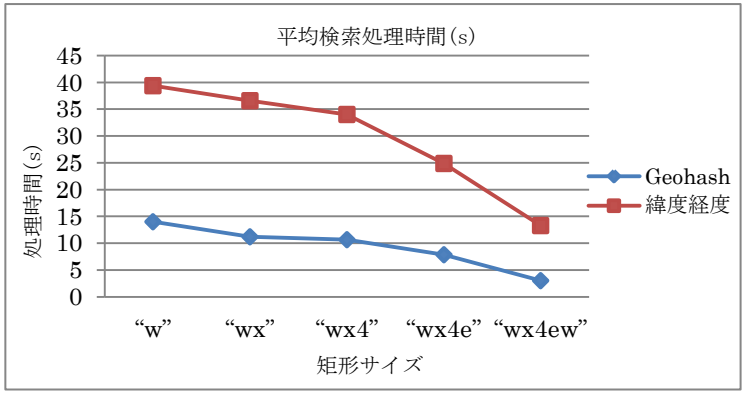

図 12　geo-string，緯度経度の検索処理時間の比較

次に，評価アプリケーションでの統計結果表示は，文字列“wx4ew”を問い合わせた結果が送られてきた XML 形式のデータを Google Map に表示させた．前方一致検索により得たデータを人口の数により色分けを行なった．今回の実装では，メッシュ内の人口数が 0 人なら灰色，1～4 人なら緑，5～9 人なら黄，10～14 人なら赤とした． 結果の表示画面を図 13 に示す.

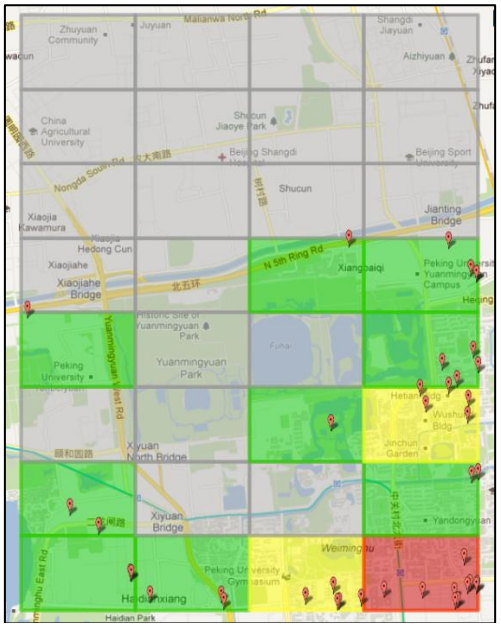

図 13 統計データの結果表示画面

6. 考察

本稿では，研究目的である「従来の手法では空間統計を把握する際の処理時間が増大」，「モバイルセンシングを利用した空間統計データ作成」の 2 つの課題を本提案システムが解決できているかの検証を行う．提案システムの評価において，従来の緯度経度による検索システムとの比較を行い，5.5 節の図 12 で示した実験結果のグラフから，統計処理の空間範囲を狭くする，つまり，アプリケーションが統計処理の範囲を絞り込むにつれて，データベースシステムの応答が速くなった．グラフが示す結果から，処理時間は減少しており，従来の緯度経度を用いた検索処理より

geohash を用いた検索処理が速度向上している．以上のことから，「従来の手法では空間統計を把握する際の処理時間が増大」の課題が解決したと言える．

また, 図 13 の結果からメッシュ毎にモバイルセンサを分類し人口数の統計データの取得ができた. これにより,「空間統計データの作成」の課題が解決したと言える. さらに, 評価アプリケーションである Google Map の表示においてもどのアプリケーションでも扱うことのできる XML 形式に変換し，統計処理をデータベースシステム側で行い結果のみをアプリケーションに送り表示させることでアプリケーションの処理の軽減に繋がっている．

以上により，本稿では研究の課題であった「空間統計を把握する際の処理時間が増大」を解決し，モバイルセンシングを利用した空間統計データの生成及び管理手法の検証を行うことができた．

しかし，検索処理において主記憶、キャッシュを用いた検索処理を加えていないのでそれらの要素を考慮する必要がある

7. おわりに

本稿では，空間統計を把握する際の処理時間が増大する課題を解決するために，階層的な空間データ構造である geohash を用いたデータベースシステムの処理速度向上及び空間統計データ生成・管理手法を提案した．そして，データベースシステムのプロトタイプシステムと空間統計データの表示に必要な評価アプリケーションの実装をして提案システムの有用性についての検討を行った．結果として，geohash を用いることで問い合わせの処理時間を減少させることができ有用であると言えた. また，空間統計においても，geohash を用いることでメッシュ状に表示することができ統計データに対しても有効であると言える．今後は，複数の種類のモバイルセンサ（車，バス，電車など）のセンサデータを考慮したデータの追加した空間統計データの生成や，それに伴う結果表示の改良を行いたいと考えている．

参考文献

[1] Goldman，J. et al. :“Participatory Sensing : A citizen-powered approach to illuminating the patterns that shape our world，” White paper published by Woodrow Wilson International Center for Scholar．September，2008.

[2] Campbell，A. T. et al. :“The Rise of People-Centric Sensing，”IEEE Internet Computing，Vol.12，Isuue 4，pp.12-21，2008.

[3] 幸田拓耶，岩本健嗣，高汐一紀，徳田英幸：“モバイルセンサノードによる環境情報収集手法”，情報処理学会研究報告．UBI，ユビキタスコンピューティングシステム，pp.17-22，2004.

[4] 松浦寛，西山裕之：“高機能携帯電話を用いたライフログ収集手法の提案及び状況推測に関する研究”, 情報処理学会, 全国大会講演論文集 2011(1), pp.291-293, 2011.

[5] 加藤大智，山岸弘幸，鈴木秀和，小中英嗣，渡邊晃：“スマートフォンとセンサを活用したリモート見守りシステムの提案”, DICOMO2011, pp. 691‐696, 2011.

[6] Sun Microsystems，“Before You Begin Sun SPOT Tutorials，” HTML， Available at “http://www.sunspotworld.com/Tutorial/index.html”.

[7] NTT ドコモ“モバイル空間統計”， HTML Available at, “http://www.nttdocomo.co.jp/corporate/disclosure/mobile_spatial_statistics/“

[8] People Centric Sensing,“ユビキタスコンピューティングシステム講座（NAIST 安本研究室）” ，HTML，Available at “http://ubi-lab.naist.jp/mobile.html”.

[9] 木實新一，石塚宏紀，岩井将行，宮崎純，瀬崎薫，戸辺義人：“I-Tree：異種センサデータの統合利用を支援する複合型検引機構”, DICOMO2010, pp.92-99, 2010.

[10] 高橋郁久，大島裕明，山本光穂，岩崎弘利，小山恥，田中克己：“Wikipedia リンク構造を用いた歴史エンティティの重要度計算”， DBSJ Journal 10(1), pp.25-30， 2011.

[11] B.Martins，N.Freire，J.Borbinha：“Complex data transfor-mations in digital libraries with spatio-tmporal information，” In: Proc. of the 11th International Conference on Asia-Pacific Digital Libraries (ICADL 2008)， pp.174-183， 2008.

[12] “緯度経度を文字列で表 Geohash” ，HTML，Available at “http://blog.masuidrive.jp/index.php/2010/01/13/Geohash/”.

[13] Google，“Google Maps API family – Google Code”，HTML Available at “http://code.google.com/intl/ja/apis/maps/index.html”.

[14] Y.Zheng，L.Zhang，X.Xie，W.Y.Ma : Mining interesting locations and travel sequences from GPS trajectories：In Proceedings of International conference on World Wild Web (WWW 2009)，pp.791-800，2009.