情報処理学会論文誌 Vol.58 No.1 2–12 (Jan. 2017)

カーネルガウシアンプロセス回帰による時空間分布データ削減方式

浅原 彰規[1,a)] 林 秀樹[1,b)]

受付日 2016年4月6日, 採録日 2016年10月4日

概要：本論文では気温や降水量など物理量の分布を表すデータベースの構築時，周辺データとの類似度の線形和で記述できる近似曲線を用い，検索の機能を損なわずにデータ数を削減する方式を提案する．従来，物理量の分布データは件数が多すぎるため，検索性能を高めるにはハードウェアなどのコストがかかりすぎるという問題があった．提案方式では，カーネルガウシアンプロセス回帰を用いて分布の近似曲線を求め，その計算に必要なデータのみを管理することでデータ件数を削減する．また，標高および降雨量の2種のデータによる実験により，検索性能を維持したままデータ件数は数%に減らせることが確認できた．これにより，低コストで物理量の分布データの検索機能を提供できると期待される．

キーワード：配列データベース，時空間データベース，回帰分析

Spatio-temporal Distribution Data Reduction Based on Kernel Gaussian Process Regression

Akinori Asahara[1,a)] Hideki Hayashi[1,b)]

Received: April 6, 2016, Accepted: October 4, 2016

Abstract: We propose a method to reduce the number of physical-quantity distribution data (i.e. temperature and rainfall) to manage it with relational database systems. Database systems for providing search functions take extremely high cost, due to requirements for hardwares. The proposed method thus takes an approach that an approximation curve function derived with kernel Gaussian process regression in advance determines the minimal dataset to be input into database systems. By results of experiments using digitized elevation map (DEM) data and rainfall distribution data, we confirmed that the proposed method could make the number of data drops down to less than 1/10 of the original data number. These results demonstrates that the proposed method is available for managing the physical-quantity-distribution data with low cost.

Keywords: array database, spatio-temporal database, regression

1. はじめに

1.1 背景

近年，IoT（Internet Of Things）とビッグデータの関連が取沙汰され始めた．IoTとはインターネットに接続したセンサから様々な計測データを収集するシステムを指す．一般的にセンサによって得られるのは，温度や加速度などの物理量や位置情報であるので，IoT時代のビッグデータ管理システムには大量の物理量のデータを扱う機能が求められる．そこで本論文では数値シミュレーションなどによって得られる，気温，降雨量などの時間的，空間的に分布するデータ（以降，分布データとする）をリレーショナルデータベースシステム（以降，RDBMS）によって扱うことに焦点を当てる．

RDBMSを用いた分布データの取扱いについて，具体例として時間帯ごとの降雨量の分布を例にあげて説明する．降雨量の分布は降雨レーダ[17]などにより継続的に計測さ

[1] 株式会社日立製作所研究開発グループシステムイノベーションセンタ
Research & Development Group, Center for Technical Innovation, Hitachi Ltd., Kokubunji, Tokyo 101–0062, Japan
[a)] akinori.asahara.bq@hitachi.com
[b)] hideki.hayashi.xu@hitachi.com

れている．このデータは空間を格子状に区切った区画（グリッド）の単位の降雨量が記述された分布データであり，日時を指定すればそのときの降雨量を参照できる．この降雨量のデータをセンサのデータと組み合わせる場合には，時間帯と地点を指定し，そこでの降雨量が特定の条件を満たす日を知りたい，ということが想定される．たとえば，ある日の午後 3 時，国分寺市の降雨量として 50 mm を観測したとき，過去の類似する日の情報，つまり「過去 10 年間で午後 3 時台に国分寺市の降雨量が 50 mm 以上であった日の気温変動が知りたい」などである．このような複合的な条件によるデータの絞り込みは，ビジネスインテリジェンスの分野ではドリルダウンという名で知られており [13]，データから新たな知識を求めるための手法としては非常に一般的なものである．このような機能は，RDBMS を用いて実装されることが多い．RDBMS は SQL と呼ばれる言語を用いて記述された条件に基づく効率的なデータの取得機能を持つため，絞り込みの条件を SQL で記述することで，比較的容易にドリルダウンが実装できる．

ところが，RDBMS を用いて分布データのドリルダウンを実装するには 1 つの問題がある．それはデータの数である．分布データの内容に対する条件で検索をするには，グリッドそれぞれを RDBMS のレコードとして格納する必要がある．この分布データのグリッドそれぞれは決して大きなデータではないが，その反面データの件数は爆発的といってよいほど多い．たとえば，日本全国（面積約 370,000km^2）の降雨量の分布を 1km^2 のグリッドで覆って表現するには，およそ 370,000 件のグリッドが必要である．これを 1 時間ごと 10 年間集めると，約 320 億件ものデータ件数となる．近年では 250 m グリッドで毎分の計測が可能な X バンド MP レーダ [11] も普及し始めており，もしそれを同様に扱うならば 30 兆件という超巨大規模の件数になってしまう．

このような多数のデータの中から高速に必要なデータを探し出す処理としては，多数のストレージに並列にアクセスする索引構造を持つ RDBMS を用いた方式 [2], [14] や，ストレージではなくメモリ上にデータを保管するプラットフォームが知られている [3]．ただし，並列アクセスにはデータを多重化しなければならず，十分大きなストレージ容量を要する．また，メモリ上に大量のデータを保持するにはそれだけ大容量のメモリが必要になってしまう．もしこれらの特殊な環境を避け，何とか一般的な RDBMS を適用できたとしても，データの管理に要する負荷（たとえば，インデックスデータや予備領域）は件数に比例して増大してしまい，要求されるハードウェア性能が高くなり，また，データのバックアップやレプリケーションなどの管理作業にも時間を要するようになる．結果的に，極端に多くのデータを高速にアクセス可能な形で管理しようとすると，コストの増大が避けられない．すなわち，RDBMS に格納するデータ件数を減らさなければ本質的に問題解決しないと考えられる．

そこで本論文では，物理量の分布が一般的には滑らかである点に着目し，回帰分析を利用したデータ件数の削減方式を提案する．ここで回帰分析とは多変量データの傾向をよく近似する曲線（回帰曲線）を求める処理であり，回帰曲線の計算に必要なデータのみ RDBMS に格納すれば，検索時に回帰曲線から元のデータを復元できる．もし，近似にともなう誤差に上限があるならば，検索条件をその分広くすることで本来の検索結果を包含する結果が得られるため，絞り込みに活用できると期待される．

提案方式では，回帰分析の方法としてカーネルガウシアンプロセス（Kernel Gaussian Process; KGP）回帰 [6] を用い，回帰分析の誤差が所定の値の範囲に収まるようにデータを選択する．KGP 回帰は新たなデータの追加をガウス分布の更新過程と見なし，回帰曲線の更新をモデル化した方式であり，その回帰曲線が周辺のデータとの類似度の線形和で記述できるため，SQL 文のみで復元処理が記述できるという利点がある．提案方式では，分布データから数点のデータを適当に抽出して回帰曲線を求め，その分布の誤差が所定値以下になるまで逐次的に新たなデータを追加して回帰曲線を更新し高精度化を繰り返す．その結果を RDBMS に格納しておき，検索時はこのデータを回帰曲線の補完処理を SQL 文に組み入れて検索する．これにより，RDBMS に投入するデータ件数を削減しつつ，検索が可能な状態を維持することができる．また本論文では，いくつかのサンプルデータを用いて提案方式の有効性を評価した結果についても報告する．

2. 従来方式の課題

2.1 時空間索引を用いた検索方式

一般的に分布データは NetCDF [8] などの形式のファイルで管理される．通常，分布データに格納すべき情報は極端には多くないので，ファイルに書かれているデータをまとめて読み出すのはさほど時間を要しない．しかし，分布データが多数ある場合に必要なものを抜き出す検索処理の実装は簡単ではない．特に検索の条件を後から自由に変えられるような柔軟性を持たせるには，RDBMS の表に分布を構成するグリッドデータそれぞれを行として格納することが望ましい．**表 1** に検索条件の柔軟性を念頭においた

表 1 分布データの格納項目
Table 1 Contents of a distribution data.

項番	カラム名	内容
1	did	分布の ID（例：日ごとの分布なら日付）
2	x	当該レコードの物理量の位置の x 座標
3	y	当該レコードの物理量の位置の y 座標
4	t	当該レコードの物理量の時刻
5	v	物理量の値を意味する浮動小数点数値

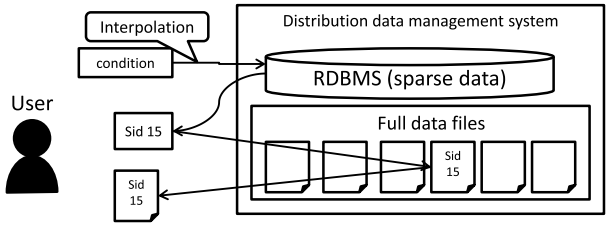


図 1 分布データ検索システムの構成
Fig. 1 Distribution-data retlieval system.

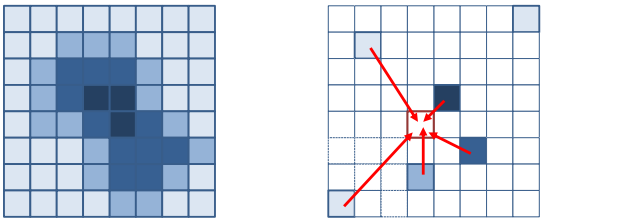
分布データの表の形式を示す．この形式の表に SQL で記述された条件を送付することで自由に検索できる．たとえば，単位が m の座標系，時刻を時間単位，日ごとに分布の ID を付与したとき

```
select did from distribution where
 (x between 10 and 20)
 and (y between 10 and 20)
 and (t between 10 and 15) and (v > 100.0)
```

という SQL 文により，位置 x，y がそれぞれ 10 m～20 m，10 m～20 m の矩形内で，時刻が 10 時～15 時の間に，物理量（たとえば降雨量）が 100.0 以上を記録したような日の ID を検索できる．しかし，この構造でデータを管理すると，データ件数が極端に多くなってしまい，条件を満たすレコードの発見に時間を要するという問題がある．これを解決する手段として，たとえば x，y，t に時空間索引を構築するという方法がある [2]．しかし，前述のとおり，データの件数が極端に多くなると管理に要する負荷（計算量や記憶容量）が大きくなってしまう．

RDBMS でのデータ管理件数を低減させる方法として，RDBMS とファイルを併用する方式が考えられる．そのシステム構成を**図 1** に示す．この方式ではファイルごとに平均値などの概略的データ，いわば目録のデータを RDBMS に格納しておき，まずそれを検索して，どのデータにアクセスするべきかを特定する．その後，当該特定ファイルからデータの実体を読み出すことで，RDBMS で管理するデータの件数を少なく保つことができる．この方法では，事前に検索の条件を想定して概略的データを作成しておく必要があるが，実際にはドリルダウンなどでの検索条件は非常にパターンが多く，事前にあらゆる条件を網羅しておけることは少ない．

そこで提案方式では，RDBMS で管理するデータとして分布データを間引いたものを用い，その間引かれた分布データに補完処理を施すことで元の分布データを高い精度で再現する方針をとる．これにより，分布データそのものに対する検索と同様の機能を提供しつつ，データ数を削減できることが期待される．



(a) 分布データ (b) データ削減後の分布データと補完処理

図 2 分布データのデータ削減の模式図
Fig. 2 Illustration of Distribution-data reduction.

2.2 カーネルガウシアンプロセス回帰を用いた分布データ削減

前述のとおり，間引いた分布データを補完することで間引く前の分布データをよく再現できるようであれば，RDBMS で管理するデータの件数を減らしつつ，管理効率を高めることができると考えられる．分布を補完する方法はスプライン補完など様々なものが知られているが，本論文では，機械学習の分野で用いられる KGP 回帰を用いた方法を提案する．KGP 回帰は，ガウス過程による線形回帰分析にカーネル法を適用することで，柔軟性の高い回帰曲線を特定する方式の 1 つである．

KGP 回帰は地球物理学分野で Kriging と呼ばれるものと同等である．Kriging はある地点の物理量はその周囲と類似するという仮定のもと，周囲のデータに関する重ね合わせによって分布を推定する方式であり，降雨量のような滑らかな分布をよく近似すると考えられる．**図 2** にこの概念図を示す．図 2 (a) が元の分布データであり，ここから分布データを構成する多数のグリッドデータから代表的な少数のサンプルを抽出したものが図 2 (b) である．図中，赤い矢印で示されているように，KGP 回帰では周囲のデータとの類似度の線形和によって任意の座標での分布の値を補完することができる．周辺検索は SQL 文で記述できるので，周辺のデータを検索してそれとの類似度に係数をかけて和の集約演算を適用するように SQL 文を記述すればそれだけで分布が補完できる．以下にその例を示す．

```
select did, sum(w*k([{x0,y0}], {x,y}))
 from distribution
 where dist([{x0,y0}],{x,y}) < [r]
group by did
```

ここで [] で囲まれた部分は定数に置き換える部分を意味する．[{x0,y0}] は検索対象となる x，y 座標，[r] は周辺検索範囲の半径のパラメータを意味する．did は分布の ID を意味しており，gourp by 句によって同一の分布内で集計することを指示している．また，w は各点の重み，k({x0,y0}, {x,y}) は指定された点 [{x0,y0}] とデータの位置 {x,y} の類似度を求める関数，dist({x0,y0}, {x,y}) は点 [{x0,y0}] とデータの位置 {x,y} の距離を求める関数である．この

SQL 文によって，全分布の点 [{x0,y0}] における近似値が計算できる．

前述のとおり，KGP 回帰は滑らかな分布を想定した近似方式であるため，仮に一律にデータを間引きしてしまうと，ゲリラ豪雨のような極端な変動を KGP 回帰の近似計算で補完することは難しい．そこで提案方式では，KGP 回帰に基づく近似値が一定の誤差 ϵ 範囲に収まることを保証しつつ，なるべくデータ数が少なくなるように間引く方針をとる．この誤差 ϵ で分布データを再現できるような間引かれたデータ集合を，以降ではこの分布データの誤差 ϵ での疎表現データセットと呼ぶ．

提案方式では疎表現データセットに対し誤差 $\pm\epsilon$ を許容して検索する．この検索は絞り込み検索として機能するので，その後，必要であればファイルで管理されるデータに直接アクセスすることで正確なデータを取得することができる．**図 3** に一定誤差を許容する検索の模式図を示す．図中縦軸 F は物理量を表し，横軸 x は x 座標を表す（簡単のため y 軸は省略した）．また，黒円は RDBMS に格納された分布データを意味し，白円は間引かれた分布データを意味する．実線で示したのが RDBMS に格納されている分布データから分布を推定した場合の関数で，$x = x_i$ のところでは，★印の値が得られる．この値は白円が示す元の分布データとは異なっているため，$x = x_i$ において $F < F_i$ を満たすかどうかの判定は誤った結果となる．しかし，誤差が ϵ 以内であるならば，近似値は元の分布データから $\pm\epsilon$ の範囲に存在するはずである．つまり，検索の範囲を ϵ だけ広げ，「$x = x_i$ において $F < F_i + \epsilon$」という条件で検索すると，その範囲に近似値が存在する．したがって，その検索結果には，少なくとも「$x = x_i$ において $F < F_i$」という条件を満たすものはすべて含まれていることが保証できる．もちろん，この検索結果には「$x = x_i$ において $F < F_i$」を満たさないものも含むことがありうる．ただし，図 1 で示したように検索のための情報を管理する目的では，該当する分布を絞り込むことができれば十分である．つまり，RDBMS による検索で得られた分布それぞれのファイルにある元の分布データを参照し，条件を満たさないものを排除すれば，検索結果を正確にすることができる．

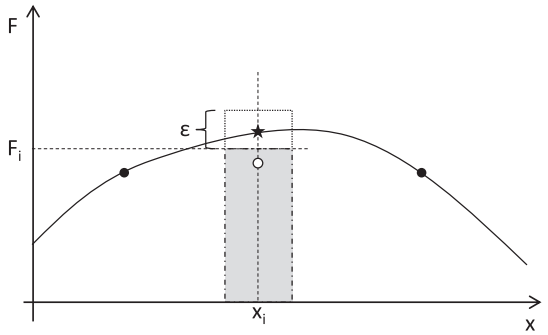


図 3 分布データ検索の条件
Fig. 3 Distribution-data retlieval condition.

3. 提案方式のアルゴリズム

3.1 疎表現データセット生成処理

前節で示した方法で，RDBMS に格納される分布データの数を減らすためには，分布データから誤差 ϵ 以内で分布を再現しうるなるべく小さな疎表現データセットを抽出する処理が必要である．提案方式では，KGP 回帰を用いてデータ件数を削減するため，ごく少ないデータを初期値としてデータを 1 つ 1 つ追加しながら KGP 回帰を繰り返し，初めて誤差の最大値が閾値以下になったときに使用したデータを疎表現データセットとして用いる．

この手続きを擬似コードにしたものを**図 4** に示す．まず初期値として非常に小さいデータセットを設定する（コード 2 行目）．最終的に得られる疎表現データセットはこの初期値によって異なる．初期値は疎表現データセットに加えられるべきかの評価がなされないため，不要であってもそのまま疎表現データセットに追加されてしまう．また周辺のデータに影響を及ぼし，さらに多くのデータを追加しなければならないことも考えられる．よって，初期値にはできるだけ少ないデータ（たとえば 1 点だけ）を用いたほ

```
1  Distribution generateSparseRepresentationDataset(Dataset originalDataset){
2          Dataset output := generateInitialData(originalDataset);
3          while (output.size()  != originalDataset.size()) do
4                  Matrix graminv := calcGramInv(output);
5                  Vector wvector := calcWeight (graminv, output);
6                  Point pt := findMaxErrorPoint(wvector, originalDataset)
7                  if (errorAt(pt, wvector) < epsilon) then break;
8                  output.append(pt);
9          end;
10         return Distribution(output, wvector);
11 }
```

図 4 疎表現データセット生成処理アルゴリズムの擬似コード
Fig. 4 Pseudo code of minimum dataset extratction algorithm.

うが疎表現データセットを小さくする効果がある．また，分布の中心点のデータ 1 つを初期値とするなどの簡易的な方法で選択してもよいが，全データの平均値との差が大きい点を選択するなど，仮に他の初期値を選択したとしても選択されたであろう点を選択することで，その影響をさらに低減できると考えられる．逆に多くのデータを初期値とすると，疎表現データセットが大きくなる代わりに KGP 回帰の繰返しが少なくなり疎表現データセット生成の処理時間は短い傾向にある．したがって時間とデータサイズのどちらを優先するかに応じて調整できる．なお，どのような初期値を選んでも誤差は $\pm\epsilon$ の範囲になるため，検索が実行できなくなることはない．次に，そのデータセットを用いて KGP 回帰分析を行う（コード 4，5 行目）．座標値 x_i とそこでの分布の値 v_i の対からなるデータセット $(x_1, v_1), (x_2, v_2), (x_3, v_3), \cdots$ に対して，計測誤差を 0 とおいた場合に，関数 $v = f(x)$ を求める KGP 回帰は以下のように計算できる．

$$v = \begin{pmatrix} v_1 & v_2 & \cdots \end{pmatrix} G^{-1} \begin{pmatrix} \kappa(x_1, x) \\ \kappa(x_2, x) \\ \vdots \end{pmatrix} \tag{1}$$

ここでグラム行列 G は

$$G = \begin{pmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{2}$$

である．この κ はカーネル関数と呼ばれる 2 つのデータの間の類似性を評価する関数であり，2 点間の距離についての多項式関数をはじめとして，様々なものが提案されている．提案方式ではカーネル関数として以下の 2 点間の距離 $|x_i - x_j|$ の関数を用いる．

$$\kappa(x_i, x_j) = \begin{cases} \left(1 - \dfrac{|x_i - x_j|}{\theta}\right)^{2n} & \left(\dfrac{|x_i - x_j|}{\theta} < 1\right) \\ 0 & \left(\dfrac{|x_i - x_j|}{\theta} > 1\right) \end{cases} \tag{3}$$

ここで，θ，n はパラメータとして与える．なお，時空間的なデータ，つまり座標が時刻を含む場合，時刻の単位を変換するための係数が必要であり，その係数も実質的に θ 同様のパラメータになる点は注意を要する．この関数は半径 θ の外側は 0 になる．したがって，

$$v = \sum_{|x_i - x| < \theta} w_i \kappa(x_i, x) \tag{4}$$

と書ける．この KGP 回帰の式は

$$v = \sum_i \left(\sum_j y_j \kappa(x_i, x_j) \right) \kappa(x_j, x) \tag{5}$$

と変形できる．このとき重み係数 w_i を

$$w_i = \sum_j y_j \kappa(x_i, x_j) \tag{6}$$

とすると，ある x での分布の値は，$\kappa(x_j, x)$ を計算して各々に係数 w_i を乗算して和をとれば求まる．すなわち，w_i を計算すれば回帰分析は完了したことになる．

次に現在使用しているデータのみを用いた場合の分布の推定誤差を評価する（コード 6 行目）．前述の方法で，現在回帰分析に使われていないデータそれぞれに対して v の推定値を計算し，元の分布データの値 v_i との差を評価する．最大誤差が $\pm\epsilon$ の範囲に収まらない場合，誤差が最大のデータを追加する．そうして回帰分析の誤差の評価を，誤差の最大値が所定の閾値以下に収まるまで繰り返す（コード 7，8 行目）．上記の KGP 回帰の式は分布データがあるところはそれと一致する値を返すため，少なくともすべてのデータを追加するまでには誤差の最大値が所定の閾値以下に収まるはずである．

この手順により，前述のゲリラ豪雨のような，局所的に大きいなどの異常値を含む分布データに対しても，必ず誤差が閾値以下であることを保証できる．実際に異常値を含む場合の挙動としては，まず異常値は回帰分析の誤差が大きい地点として現れるため，優先的に回帰分析に使用されるデータとして選択されると考えられる．すると，KGP 回帰は滑らかな分布を仮定するため，この異常値付近も同様に大きな値が入るという推定結果になるはずであり，異常値付近の異常値でない地点について，回帰分析の誤差が大きい，という結果が得られると考えられる．したがって，回帰分析の誤差が十分小さくなるまで，異常値周辺のデータが回帰分析に使用されるデータとして選択される．つまり，異常値があっても異常値自身とそれを取り囲むようにデータが使用され，いずれも誤差が閾値以下であることを保証できる．

最終的に得られた重み係数 w_i と座標値 x_i を RDBMS に格納すれば，以下のような周辺検索を含む SQL 文によって地点 [{x0,y0}] の物理量が誤差 ϵ 以内で再現できる．

```
select did, sum(w*
pow(1-dist([{x0,y0}],{x,y})/[theta],[2n]))
from distribution
where dist([{x0,y0}],{x,y}) < [theta]
group by did
```

ここで [{x0,y0}] は検索対象となる x，y 座標，[theta] はカーネル関数のパラメータ θ，did は分布の ID，w は各点の重み w_i，dist({x0,y0}, {x,y}) は点 [{x0,y0}] とデータの位置 {x,y} の距離を求める関数をそれぞれ意味する．なお，空間索引などにより周辺検索の部分は高速化できる．

3.2 アルゴリズムの高速化

KGP 回帰においては，一般的にグラム行列の逆行列 G^{-1}

を求める計算量の大きさが問題になることが知られている．実際，その点は提案方式においても処理時間がかかる要因となる．そこで，提案方式ではこれを高速化するために逆行列を逐次更新するようにする．

提案方式のアルゴリズムでは，逐次的にデータを追加することになるため，初回を除きグラム行列の逆行列について，データ数が少ないときの計算結果がすでに存在する．これを利用することで，毎回すべての逆行列を計算するのに比べて，計算量を大きく削減できる．n 番目までのデータに対するグラム行列を G_n としたとき，$n+1$ 番目のグラム行列 G_{n+1} は以下のように書ける．

$$G_{n+1} = \begin{pmatrix} G_n & c_{n+1} \\ c_{n+1}^T & \kappa(x_{n+1}, x_{n+1}) \end{pmatrix} \tag{7}$$

ここで

$$c_n = \begin{pmatrix} \kappa(x_1, x_n) \\ \vdots \end{pmatrix} \tag{8}$$

とした．ここで区分行列の逆行列の公式を用いると，

$$G_{n+1}^{-1} = \frac{1}{s} \begin{pmatrix} sG_n^{-1} + VV^t & -V \\ -V^t & 1 \end{pmatrix} \tag{9}$$

と変形できる．ここで，

$$s = \kappa(x_{n+1}, x_{n+1}) - c_{n+1}^T G_n^{-1} c_{n+1} \tag{10}$$

$$V = G_n^{-1} c_{n+1} \tag{11}$$

とおいた．G_n^{-1} と c_{n+1} と $\kappa(x_n, x_n)$ だけを用いて G_{n+1}^{-1} が逐次的に計算できる．

上記により逆行列の計算は高速化されるが，ほかにもう1つ大きな処理時間を要するものがある．それは，誤差が最大のデータを選択する処理である．この処理ではすべてのデータに対して近似関数を評価する必要があるためである．ただし，カーネル関数の計算が波及範囲を勘案すると計算量が削減できる．提案方式で用いたカーネル関数は半径 θ の外では 0 であるため，新しいデータが追加されても距離が θ 以上の範囲の補完結果には影響しない．そこで，推定値を毎回計算するのではなく，計算した結果を保存しておき，新たに追加されたデータの周辺のみ更新すればよい．

このアルゴリズムでは，カーネル関数のパラメータ θ によって計算すべき範囲が決定されるため，θ が大きいほど G_{n+1}^{-1} の逐次計算の処理時間がかかるようになる．その反面，θ が大きいほど1つのデータが影響を与える領域が広くなるため，滑らかな分布を表現する場合には θ が大きいほうが疎表現データセットのデータ件数を小さくできる傾向にある．たとえば θ が極端に大きい場合，$\kappa(x_j, x) \neq 1$ となり，その場合は1点のみデータを用いても式 (1) が一様分布 $v = v_1$ となるので，$\pm\epsilon$ 以下の変動しかないような極端に滑らかな分布に対して，疎表現データセットの件数が1になる．この場合，逐次計算の処理時間の増大よりも疎表現データセットが小さいことによる高速化のほうが勝り，かつ疎表現データセットがより小さくできるという点で望ましい．ただし実際には，物理量の分布には滑らかでない部分，たとえば異常値もありうる．その場合に θ を大きくすると，たとえば全域を異常値と推定してしまい，打ち消すために疎表現データセットに多くのデータが必要になるなど，性能面で悪影響がある．ただし，通常，グリッドの大きさは物理量の分布の変動が適度に表現できるよう設定される．したがって，たとえば $\theta < 1$ のような極端な値は避け，可能であれば，θ は物理量の分布の滑らかさにあわせ，計算環境の性能や疎表現データセットの大きさを加味して，性能を確認しながら調整するのが望ましい．

以上によって，推定の計算にかかる計算量を低減でき，処理時間を高速化することができ，現実的な処理時間で疎表現データセットが構築できる．

4. 実験

4.1 実験環境

提案方式の有効性を検証するため，2種類のデータに対し疎表現データセットを構築する実験を行った．今回，空間的な分布を表現するデータと，時空間的な分布を表現するデータの2種を対象に，それぞれ実験1，実験2として性能を評価した．実験に用いた環境は OS が Windows 7，CPU は Intel Corei-7 3770K 3.5 GHz，メモリ 16 GB のコンピュータであり，アルゴリズムの実装には Java を用いて実装し，JDK 1.8.0update31 を用いて動作させた．また，データを格納する RDBMS としては PostgreSQL 9.2 および PostGIS2.0 を用いた．

4.2 実験1：DEM データによる性能評価

実験1では地形の標高を計測したデータである DEM（Digitized Elevation Map）を用いた．日本国内の DEM データは航空レーザ測量などを用いて計測されたものが公開されている．今回，国土数値情報 [15] の「標高・傾斜度5次メッシュデータ」を用いた．**表 2** に DEM データの概要を示す．このデータには，当該エリアを 320×320 に分割したグリッド（幅はおおむね 250 m）の単位で，海中以外の地表面の高さが格納されている．全体は 102,400 件のデー

表 2 DEM データの概略
Table 2 Overview of the DEM dataset.

項目名	内容
データサイズ	RDBMS 上で 3.8 MB
データ件数	全データ 102,400 件 地表面のみ 74,663 件
地域	1次メッシュ番号 5238 の地域（静岡県周辺）

表 3 実験 1 評価結果概略
Table 3 Results of the 1st experiment.

項目名	元データ（地表のみ）	提案方式（ϵ =125）	提案方式（ϵ =250））
テーブルデータサイズ	3.8 MB	152 kB	64 kB
インデックスデータサイズ	3.0 MB	168 kB	64 kB
データ件数（全数比）	72.9%	2.8%	1.1%
生成処理時間	なし	383.8 s	14.6 s
検索時間（100 レコード）	21 ms	110 ms	51 ms

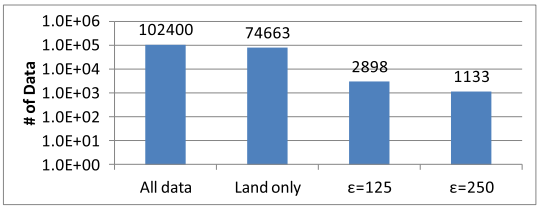


図 5 DEM に関するデータ数削減効果
Fig. 5 Data reduction for the DEM dataset.

タに相当するが，海中のデータを除外すると 74,663 件のデータであった．なお，当該テーブルでは，x，y の 2 つについてマルチカラムの索引を構築した．今回，標高 250 m 程度を 1 つの区切りとして扱うことに配慮し，まず 125 m の誤差を許容して疎表現データセットを構築した．カーネル関数のパラメータは，グリッド 1 つ分を座標の単位として $\theta = 5$（およそ 1,250 m），$n = 2$ とした．次いで同一のパラメータで 250 m の誤差を許容した場合の疎表現データセットを構築し，その結果を比較した．

表 3 に実験 1 の結果をまとめたものを示す．この中で特に構築された疎表現データセットと元の DEM データの定量的な比較を**図 5** にグラフの形で示す．本実験では，全数 102,400 件，地表のみ（Land only）74,663 件であったデータは提案方式により 2,898 件まで削減された．つまりデータ件数はおよそ 3.9%に減ったことになる．このとき，疎表現データセットを求める処理には 383.8 秒を要した．許容誤差を 250 m にした場合は，データ件数は 1,133 件，つまり，1.5%にまで削減できており，許容誤差を大きくとればそれだけデータ数が減らせることが分かる．このときの疎表現データセットを求める処理時間は 14.6 秒であった．また，データ件数の削減にともない，RDBMS のテーブルデータサイズ，インデックスデータサイズも減少した．よって，提案方式はデータサイズを大きく削減できる方式であり，検索範囲が広くなるほど復元に計算時間がかかるものの，対象が少ない場合は元データを検索するのと大差ないことが確認できた．

また，疎表現データセットをもとに SQL による検索性能を評価した．$\epsilon = 125$，$\epsilon = 250$ の疎表現データセットに対して，100 グリッドの値を求める SQL 文を実行したところ，$\epsilon = 125$ では 110 ms，$\epsilon = 250$ では 51 ms で計算が終了した．他方，元のデータセットに対し，索引を構築した状態で同じグリッドに対する検索要求をかけたところ，21 ms で結果が得られた．**表 4** に検索結果の比較を示す．検索結果は誤差を含むものの，ϵ で指定された許容誤差以下であることが確認できる．また，ϵ が異なる場合でも，推定値は大差ないことが確認できる．ϵ は疎集合データセットへのデータ追加の終了判定にしか用いられないため，誤差が多くないところではほぼ同じ推定が用いられるためである．この性能評価においては，疎表現データセットを用いたほうが処理時間が大きくなった．とはいえ，ドリルダウンなどの用途では分布全体のデータを取得するような検索は考えがたいため，この処理時間は実用上問題があるほどではないと考えられる．一方で，データサイズは大きく削減できており，提案方式の有効性が確認できた．

表 4 DEM の検索結果の例
Table 4 Results of retrieving the DEM dataset.

提案方式（$\epsilon = 125$）	提案方式（$\epsilon = 250$）	元の分布データ
621.3	621.3	601.7
551.0773214	533.6383138	542.8
572.9943385	589.7878391	611.2
553.5379556	559.0510806	514.8

表 5 降雨データの概略
Table 5 Overview of the rainfall dataset.

項目名	内容
データサイズ	CSV 形式 88.2 GB RDBMS 上で 142 GB
データ件数	全体 2,322,432,000 件 0 でないもの 386,685,916 件
地域	日本全国および近海
時期	2013/10/1～2014/11/4 の 400 日分

4.3 実験 2：降雨データによる性能評価

実験 2 では時空間分布データに対する効果を評価するため，時系列の降雨データを用いた評価を行った．**表 5** に本データの概要を示す．今回用いたのは，京都大学生存圏研究所の公開しているグローバル大気観測データ [12] を蓄積したデータセットである．このデータは 480×504 のグリッドで降雨量が 1 時間ごとに格納されており，データ件数はおよそ 23 億件になる．このうち，実際に降雨が起きているデータは 336,685,916 件あった．このデータは NetCDF

表 6 実験 2 評価結果概略
Table 6 Results of the 2nd experiment.

項目名	元データ（降雨のみ）	提案方式
テーブルデータサイズ	25 GB	27 MB
インデックスデータサイズ	15 GB	16 MB
データ件数（全数比）	16.7%	0.01%
生成処理時間	なし	27 hour
検索時間（100 レコード）	2,000 ms	1,700 ms

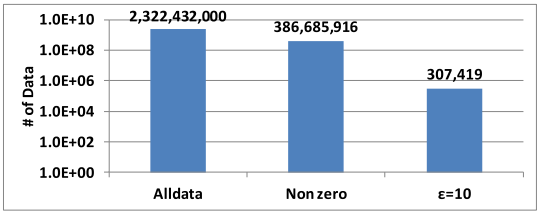


図 6 降雨データの削減効果
Fig. 6 Data reduction for the rainfall dataset.

形式であるが，RDBMS に格納するため，1 度 CSV テキスト形式に変換した．この CSV は表 1 で示したのと同じカラム構成を用いており，分布 ID did としては変換時に各日付ごとに一意な連番を付与して用いた．これは本データを日ごとの降雨量の時空間分布，すなわち空間座標 x, y とその日の 0:00 からの経過時間 t（$0 < t < 23$）の 3 変数関数と見なしたことに相当する．この CSV ファイルのデータサイズは 88.2 GB であった．それを実際に RDBMS に格納したところ，データ型などに integer 型などの CSV 内の文字列に比べてサイズの小さい格納形式を選んだにもかかわらず，テーブルサイズとしては 142 GB と CSV の 1.5 倍以上になった．ここから RDBMS に小さなデータを大量に格納することの効率の悪さがうかがえる．また，降雨の起こっていないデータを除いたテーブルを作り，そこに t，x，y のマルチカラムの索引を構築したところ，そのサイズは実体が 25 GB，索引が 15 GB であった．このテーブルに検索をかけることにより，ある時間帯，地域の降雨量についての条件（たとえば $0 < x < 10$，$0 < y < 10$，$15 < t < 16$ の平均降雨量が 50 m 以下など）を満たす did，つまり日付を特定することができる．

このデータに対し，許容誤差として降雨量 10 mm を設定して疎表現データセットを作成し性能を評価した．このときカーネル関数のパラメータは $\theta = 20$，$n = 2$ としており，時刻については時間を 10 倍したものを空間データの 1 グリッドに相当するとした．**表 6** に概略を示し，特にデータの件数については**図 6** にグラフを示す．疎表現データセットのデータ件数は 307,419 件となり，全数比でおよそ 1/1000 までデータ数が削減できた．また，RDBMS のテーブルサイズもそれに従い，小さくなっていることが確認できた．

また，疎表現データセットをもとに SQL によって復元する性能を評価するため，10×10 グリッド分の 15 時または 16 時の範囲に対して，50 mm 以上の降雨が起きた日を検索した（なお，降雨量 10 mm の誤差を許容するため，条件式は 40 mm 以上とした）ところ，およそ 1.7 s で計算が終了した．他方，元のデータセットを RDBMS に格納して，同様の検索要求をかけたところ，2.0 s 程度で計算が終了した．すなわち，検索性能としては大きな速度低下を招くことなく，データサイズを削減できたといえる．**表 7** にそのときの検索結果を示す．この結果は，上記の検索結果に対して did ごとに降雨量の最大値をとったものである．提案方式は did が 174，240，254 の 3 つ，元の分布データに対しては 174，254 の 2 つが検索結果として得られた．did が 240 の分布データについて，同じ 10×10 グリッド内の 15 時または 16 時の範囲の最大値は「43.688068182665」であるため，「50 mm 以上の降雨が起きた日」という条件から元データの検索結果には含まれていないが，許容誤差 ϵ が 10 mm である提案方式の検索結果には含まれている．このように，許容誤差の分だけ検索条件が広がり，提案方式の検索結果が元データを包含する形になったことが確認できる．この真値と提案方式の結果を比較すると近似性能は十分であったと分かり，元データと提案方式の検索結果に差異が生じているものの，それは想定された範囲であった．

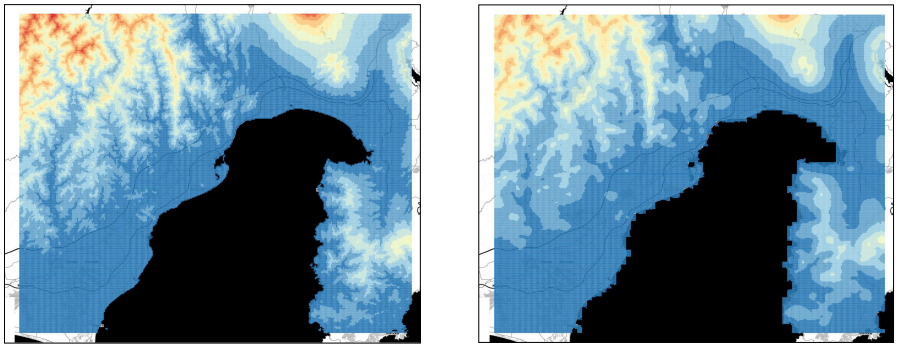
表 7 検索結果の例
Table 7 Results of retrieving the rainfall dataset.

did	提案方式	元の分布データ
174	91.3486411944194	91.31498108565
240	43.6964216739264	
254	58.5002035772889	58.489291917765

4.4 考察

上述のとおり，DEM では検索に要する時間は，少数のデータに対するアクセスでは差がほとんどみられなかった．しかしながら，疎表現データセットから元のデータを完全に復元するような処理には大きな時間がかかると想定される．そこで，実際に復元処理を行った．**図 7** に元の DEM データと疎表現データセットから再構成された近似の DEM データを可視化した画像を示す．画像中のオレンジは高度が高く，青は低く，黒は海である．また**図 8** に疎表現データセットの位置を描画した画像を示す．図中赤△が疎表現データセットの位置であり，海岸線付近や山中など凹凸が多いところが重点的に選択されているが，それでも微細な凹凸は再現されていない．この疎表現データセットからすべての DEM を再構成するのにかかった時間は，約 219 秒であった．つまり想定されたとおり，データ全体を再現しようとするとかなりの処理時間を要していることが確認できた．

同様に処理性能を評価するため，**図 9** (a)，(b) に実験 1，



(a) 元データ
(a) Original data

(b) 疎表現データセットから再構成されたデータ
(b) Data reconstructed with minimum dataset

図 7 DEM データ
Fig. 7 DEM data comparison.

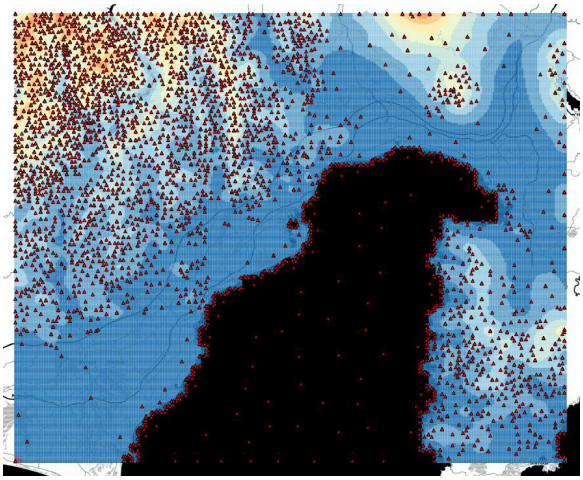
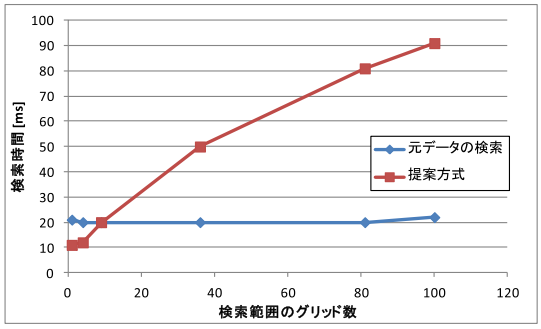
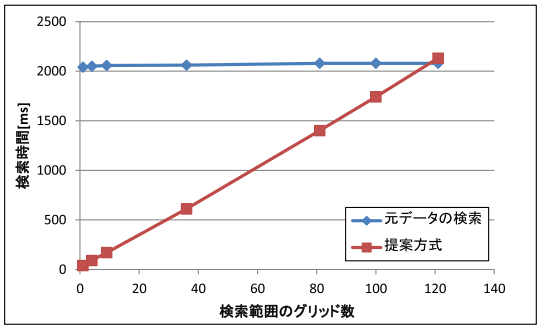


図 8 DEM に関する疎表現データセット
Fig. 8 Minimun datset of DEM.



(a) DEM データ



(b) 降雨量データ

図 9 検索範囲ごとの処理時間
Fig. 9 Processing time by retlieval ranges.

2 の条件で検索範囲を変更しながら検索した結果をそれぞれ示す．横軸は検索対象となる空間のグリッド数，縦軸はその検索にかかった時間である．元データに対する検索では範囲によらずほぼ一定の時間で処理ができるのに対し，提案方式は検索範囲が広くなるとそれに従ってほぼ線形に処理時間が増えている．元データの検索時間に対してはストレージへのアクセス時間が支配的であり，今回，空間のグリッド数を増やす際には隣接するグリッドを対象としたため，グリッド数が増加してもストレージへのアクセス回数を増やすことなく検索ができていたため，元データの検索時間は一定になっているものと想定される．一方，提案方式では復元処理の寄与が大きいため，件数が多くなるほど計算量が多くなり，処理時間が増大するという傾向を示している．本実験の環境では，実験 1 では 3×3 グリッド，実験 2 では 11×11 グリッドで検索時間が同等程度であるが，環境によって傾向は異なると想定できる．

このように，提案方式は分布データの検索範囲が広くなると計算量が増え，処理時間もかかるようになる．ただし，提案方式は分布データの検索可能性を維持したままデータ容量を削減するのが目的であり，ドリルダウンやセンサデータとの照合のような用途では，あまり広い範囲の分布データが必要とされるとは考えにくく，代わりに検索時の速い応答速度が求められる．もし，多少時間をかけてでも広域のデータを取得する必要がある場合には，Hadoop のようなバッチ処理を想定した分散処理システムが適するなど，用途に応じた使い分けが重要である．

また，他の提案方式の課題の 1 つが，KGP 回帰にかかる処理時間である．実際には，降雨データではこの処理におよそ 27 時間を要した．ただし，従来の間引きを行わないテーブル構築でも，累計でおよそ 8 時間の処理時間を要しており，提案方式で間引いた後のデータに対するテーブ

ル構築処理は数分で終了したことも勘案すれば，総合的にはデータベース構築にかかる時間は 3 倍程度になったと見ることができる．実際には，データベースのバックアップや環境の移行など運用していくうえでの様々な処置を要する場合が多く，そのたびに 8 時間程度の処理時間を要するようであれば，提案方式で間引いた後のデータを取り扱ったほうが効率的とも考えられる．実際にこのようなデータベースを構築する際には，このような運用上の負荷まで勘案して方式を決定することが肝要である．

5. 関連研究

分布のデータの検索機能を提供する手段としては，分布のグリッドそれぞれを位置情報と属性の組として扱うことで，時空間データベース [5], [7], [19] を用いて管理することもできる．ただし，上述のとおり，これらを用いてもデータ件数が多いことにともなう負荷は避けられない．

また，分布のデータ管理として配列志向のデータベースに関する取り組みとしては SciDB [10] や rasdaman [1] が知られている．これらは内部で分布データを区画に分割して管理しており，区画単位で検索を行うことでデータ件数の増大を抑止する方式をとっている．これらの配列データへのアクセスについては，ISO で Array SQL という SQL の拡張が提案されている [4]．

本論文では，分布の近似方法として KGP 回帰を用いたが，空間的に分布する大規模データに対する統計解析手法 [16], [20] は様々なものがある．ほかにも類似の方法は知られており [9]，これらのうちいくつかは提案方式と同様の目的に適用可能と考えられる．

6. おわりに

本論文では，時空間的に分布する物理量のデータに対してドリルダウンなどの検索機能を提供することを目的に，リレーショナルデータベースシステムで管理する方法について述べた．今回は，検索可能な形で分布のデータを管理しようとすると極端にデータ件数が多くなってしまうことが，検索性能の低下やバックアップなどの困難を生じるという問題について焦点を当て，分布データをカーネルガウシアンプロセス回帰によって近似することでデータ件数を削減する方式を提案した．これにより，データの件数を削減しつつも，近似の誤差を所定範囲内にとどめることができ，検索性能自体は低下させることなく取扱いの効率性を改善することができると考えられる．

また，提案方式について，地形の高さデータと降雨量のデータを用いて実験的に評価したところ，データ件数が少なくとも 1/10 程度まで削減できた．この削減したデータに対して，ある特定の地点のデータを補完する SQL 文を実行したところ，全データを格納した場合と大差ない速度でデータが取得できることも確認できた．したがって，「過去の大量の降雨量分布データを参照してある条件を満たす日のみを選択する」などの典型的な分析に必要な検索機能を RDBMS を用いて実装するにあたって，より少ないデータ容量で同機能を提供できるものと考えられる．

ただし，提案方式にはデータの削減のための処理に時間がかかるという課題がある．本論文でもいくつかの方法を提案したが，アルゴリズムのさらなる効率改善や，並列分散処理を活用するなどの改善の余地は大きいと考えられる．また，Kriging は様々な改善提案がなされている [18]．より良い近似曲線を得ることができればさらなるデータ削減につながると期待される．一方，今回対象としたドリルダウンの類の検索以外にも，たとえば「周囲 xxm の平均値を計算したい」などの統計処理的な要求も考えられ，提案方式と類似の方法で統計値に関する近似検索も考えられる．これらの改善については今後の課題である．

参考文献

[1] Baumann, P., Dehmel, A., Furtado, P., Ritsch, R. and Widmann, N.: Spatio-temporal retrieval with RasDaMan, *VLDB*, pp.746–749 (1999).

[2] Hayashi, H., Asahara, A., Sugaya, N., Ogawa, Y. and Tomita, H.: Spatio-temporal similarity search method for disaster estimation, *2015 IEEE International Conference on Big Data* (*Big Data*), pp.2462–2469, IEEE (2015).

[3] Haynes, D., Ray, S., Manson, S.M. and Soni, A.: High performance analysis of big spatial data, *2015 IEEE International Conference on Big Data* (*Big Data*), pp.1953–1957, IEEE (2015).

[4] ISO: ISO/IEC CD 9075-15 Information technology – Database languages – SQL – Part 15: Multi dimensional arrays, available from ⟨http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67382⟩.

[5] Iwerks, G.S., Samet, H. and Smith, K.P.: Maintenance of k-nn and spatial join queries on continuously moving points, *ACM Trans. Database Systems* (*TODS*), Vol.31, No.2, pp.485–536 (2006).

[6] Shawe-Taylor, J.（著），大北 剛（訳）：カーネル法によるパターン解析，共立出版 (1992).

[7] Koubarakis, M., Sellis, T., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N., Manolopoulos, Y., Nardelli, E., Pernici, B., et al.: *Spatio-temporal databases: The CHOROCHRONOS approach*, Vol.2520, Springer (2003).

[8] Open Geospatial Consortium: OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0 (10-090r3), available from ⟨http://www.opengeospatial.org/standards/netcdf⟩.

[9] Simonoff, J.S.: *Smoothing methods in statistics*, Springer (1996).

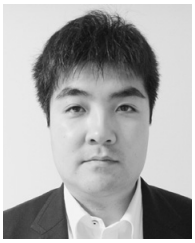
[10] Stonebraker, M., Duggan, J., Battle, L. and Papaemmanouil, O.: SciDB DBMS Research at M.I.T, *IEEE Data Eng. Bull.*, Vol.36, No.4, pp.21–30 (2013).

[11] 加藤 敦，真木雅之，岩波 越，三隅良平，前坂 剛：X バンドマルチパラメータレーダ情報と気象庁レーダ情報を用いた降水ナウキャスト，水文・水資源学会誌，Vol.22, No.5, pp.372–385 (2009).

[12] 京都大学生存圏研究所：生存圏データベース，入手先 ⟨http://database.rish.kyoto-u.ac.jp/⟩.
[13] 高間康史，山田隆志：時空間的動向情報の探索的分析を支援するインタラクティブな情報可視化システム，人工知能学会論文誌，Vol.25, No.1, pp.58–67 (2010).
[14] 合田和生，豊田正史，喜連川優：アウトオブオーダ型データベースエンジン OoODE の試作とその実行挙動，第 5 回データ工学と情報マネジメントに関するフォーラム，F3-1 (2013).
[15] 国土地理院：国土数値情報　ダウンロードサービス，入手先 ⟨http://nlftp.mlit.go.jp/AR-HMM/index.html⟩.
[16] 松田安昌：一般化 Whittle 法による不等間隔時空間データの分析，統計数理，Vol.60, No.1, pp.159–171 (2012).
[17] 深見親雄，新部明郎：全国合成レーダ雨量の精度検証，水文・水資源学会研究発表会要旨集，Vol.17, No.0, pp.130–131 (2004).
[18] 村上大輔，堤　盛人：Kriging を用いた実用的な面補間法，GIS-理論と応用，Vol.19, No.2, pp.59–69 (2011).
[19] 堀之口浩征，黒木　進，牧之内顕文：時空間データベースインデックス正規化 R*-tree の実装と性能テスト，情報処理学会論文誌，Vol.40, No.3, pp.1225–1235 (1999).
[20] 矢島美寛，平野敏弘：時空間大規模データに対する統計的解析法，統計数理，Vol.60, No.1, pp.57–71 (2012).

淺原 彰規　（正会員）

2002 年北海道大学理学部物理学科卒業．2004 年北海道大学大学院理学研究科物理学専攻修士課程修了．同年（株）日立製作所入社，以来，研究開発グループにて空間情報システムの研究に従事．電子情報通信学会員．



林 秀樹　（正会員）

2002 年大阪大学工学部電子情報通信エネルギー工学科卒業．2004 年大阪大学大学院情報科学研究科博士前期課程修了．2006 年同大学院情報科学研究科博士後期課程修了．同年（株）日立製作所入社，以来，研究開発グループにて空間情報システムの研究に従事．博士（情報科学）．ACM，電子情報通信学会，日本データベース学会各会員．