

階層的センサネットワークのための 効率的なデータ収集手法

岸野 泰 恵^{†1} 櫻井 保 志^{†1} 亀井 剛 次^{†2}
前川 卓 也^{†1} 柳 沢 豊^{†1} 岡 留 剛^{†3}

本論文では、センサネットワークにおいて、階層的な木構造のネットワークトポロジにより、効率良くセンサデータを収集する手法を提案する。提案手法では、センサデータの周期性、および近接して配置されたセンサノードで発生するセンサデータの相関性を利用し、特異値分解を用いてセンサデータを集約する。さらに階層的にこの処理を繰り返すことで、無線通信のデータ量を抑え、少ない誤差で効率良くデータを収集できる。評価実験では、階層的トポロジと特異値分解によって、階層が増加してもデータ量や誤差が低く抑えられることを示した。

Efficient Data Gathering for Hierarchical Sensor Networks

YASUE KISHINO,^{†1} YASUSHI SAKURAI,^{†1} KOJI KAMEI,^{†2}
TAKUYA MAEKAWA,^{†1} YUTAKA YANAGISAWA^{†1}
and TAKESHI OKADOME^{†3}

In this paper we propose an efficient data gathering method using hierarchical tree topology in high-density sensor network. The proposed method gathers sensor data using Singular Value Decomposition (SVD) for each cluster, by taking advantage of periodicity and correlation among sensor data. It can reduce the amount of data on wireless communication and error, and achieve efficient data gathering. Our experimental result shows hierarchical network topology and data gathering by SVD can reduce the amount of data and error when the level of network topology is high.

1. はじめに

近年、多数のセンサノードからデータを収集するセンサネットワークに関する研究がさかんに行われ^{(4),(7),(12)}、環境モニタリング、遠隔監視、科学実験、ライフログなどさまざまな応用が実現可能になりつつある。このような状況では、センサノードはさまざまな場所に埋め込まれ、光、温度、湿度、加速度、照度、ガス濃度、風力などさまざまな現象をセンシングする^{(1),(8)}。その結果は基地局へ向けて送信され、基地局はアプリケーションの要求に応じて収集したデータの分析や処理を行う。

さらに最近では、多数のセンサノードを同時に用いて、高精細な監視や、詳細な分析などを行うより高度なアプリケーションに関心が高まっている^{(9),(13)}。こういったアプリケーションを実現するためには、より効率的なデータ収集手法が必要となる。本研究では、(1) 高速にデータを収集でき、(2) 収集したデータは正確であり、(3) センサノードが低消費エネルギーで稼働できるような手法を効率的と考える。逆に、(1) データ収集の遅延が大きく、(2) 収集したデータが大きな欠損や誤差を含み、(3) 無線通信において冗長なデータの送受信やデータの再送などによって電池を大きく消耗してしまうようなデータ収集手法は効率が悪いといえる。本論文では、以上の3つの要件をバランス良く満たす効率的なデータ収集手法を提案する。

多数のセンサノードからなるセンサネットワークにおいてデータを効率的に収集するためには、階層的なクラスタを用いた木構造のネットワークが以下のような理由で有効である。

- 木構造のネットワークトポロジを採用すると、センサノードの数が増えたときにスケラビリティが高い。
- 隣接するセンサノードでクラスタを生成すると、クラスタ内の通信は微弱な電波を送信するだけで十分に通信が成立するため、電力の消費量を抑えられる。
- クラスタ内の通信は送信電力を抑えるなど、無線通信範囲を適切に調整することによって、電波が衝突する可能性を低減させることができる。このように電波の衝突による再送を防ぎ、それぞれのセンサノードの通信機会が増加することにより、効率良く通信で

^{†1} NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

^{†2} ATR 知能ロボティクス研究所
ATR Intelligent Robotics and Communication Laboratories

^{†3} 関西学院大学
Kwansei Gakuin University

きるようになる。

- 基地局の無線通信帯域には限りがあり、これが通信速度を上げようとする際のボトルネックになることが多い。階層的な木構造のネットワークトポロジを用いると階層的にセンサデータが集約されていくため、最上位階層のセンサノードと基地局との通信では、複数のセンサノードのデータが集約され、通信のオーバーヘッドが少ない。このため、基地局が1台1台のセンサノードから直接データを収集するような単純なデータ収集と比較して効率的にデータを集約できるようになる。

一方で、センサノードが計測して取得するセンサデータには以下のような特徴がある。

センサノード間のセンサデータの類似性：近接して設置されたセンサノード間では、発生したセンサデータの系列は高い類似性を持つ。たとえば温度やガスは壁などで空間が区切られない限りはセンサデータはほぼ一様に分布し、物の動きを加速度センサで測定する場合には同じ物についているセンサからは相関のあるデータが発生する。複数のセンサノードからのデータに類似性があるのであれば、類似の部分をもとめることでデータ量を大幅に削減できる。

センサデータの周期性：センサデータには周期性がある。温度や湿度は1日の周期で変化し、環境に設置したセンサで人の動作を計測するときには人が歩くなど同じ動作を繰り返せば周期性のある変化をする。周期的にデータが繰り返されるのであれば、繰り返される部分をまとめて圧縮することで効率の良い通信が実現できる。

このように、センサデータの特徴を活用してデータの集約が行えればセンサデータの量を劇的に削減できる可能性がある。

センサデータの特徴を活かしたデータ処理に関しては、これまでもさまざまなアプローチの研究が行われている。文献3)では、複数のセンサデータ間の類似性を活かして、データ量を削減する手法が提案されている。回帰分析の技術を用いて、基底となる信号に対する係数値で新たに発生したデータを表現し、データの大幅な圧縮を図っている。しかし、ネットワークのトポロジに関しては議論されておらず、階層的なネットワークにおける多数のセンサノードからのデータ集約に有効であるかは不明である。ほかに、文献5)では、センサデータに関連性があるセンサノードのデータが同じ経路上で収集されるようにネットワークトポロジを構成する手法が提案されている。関連性が非常に高いセンサデータばかりであれば、効率良くデータ収集が行えるが、さまざまな類似したデータが存在するときのようにデータを圧縮すべきであるかという点については議論されていない。あるいは、文献2)では、階層的なネットワーク上で、データをSUM, AVG, COUNT, MIN, MIX

といった関数に要約する手法が提案されている。ネットワークの階層性を活かして、大幅にデータ量を削減して、要約のためのセンサデータを収集できるが、対応する関数以外のデータを必要とするアプリケーションに対応できるかは明らかではない。

これに対して本研究では、センサデータの周期性とセンサノード間の類似性、階層的ネットワークの特徴を活かした効率的なデータ収集手法を提案する。データを圧縮する手法としては特異値分解を用いる。後述するように特異値分解を用いると、センサの時系列データを代表的なデータ系列(基底系列)とそれに対する重みの係数値に分解できる。周期性のあるセンサデータでは、類似したデータ系列が周期的に発生するため、少ない数の基底系列でも十分にデータを近似できる。また、階層的なネットワークトポロジでは、近隣のセンサノードで類似したデータ系列が発生しているため、クラスタ内でデータを集約する際に、効率良くデータ量を削減できる。このような特異値分解によるデータ集約を階層的に繰り返すと、最終的に基地局には全センサノードで共通の基底系列とそれに対する各センサノードのデータを復元するための係数値が集まる。このようにセンサノード間のデータの類似性とセンサデータの周期性を利用した近似によって、データ量を大きく削減できる。一般的にセンサノードにおいてはデータ処理の計算には無線通信に比べてわずかな電力しか消費しないため、データ量を大きく削減できる提案手法は、効率の良いデータ収集を実現できる。

以下本論文では、2章で想定する階層的なセンサネットワークについて述べ、3章で特異値分解を用いたデータ圧縮の基本的なアイデアについて述べ、4章で階層的に提案する手法を用いたデータ収集について述べる。5章でシミュレータによる評価実験の結果について報告し、6章で本論文をまとめる。

2. 階層的センサネットワーク

階層的なネットワークトポロジのセンサネットワークでは、センサノードの数が増えても効率的にデータを収集できる。図1に階層的センサネットワークのイメージ図を示す。まず最下位の階層では、各ノードは近隣に配置された他のセンサノードとクラスタを生成し、その中の1台(親ノード)へデータを集める。クラスタ内のその他のセンサノードは子ノードとなる。親ノードは他の親ノードと1段上位階層のクラスタを生成する。このクラスタでさらにデータを集約し、さらに上位階層のクラスタでデータを集約することを繰り返し、最終的には最上位の階層にある基地局にすべてのデータが集約される。階層の高さはセンサノードの数を N とするとき、 $O(\log N)$ となり、センサノードが多数ある場合でもそれほど多段のネットワーク構成になるわけではない。

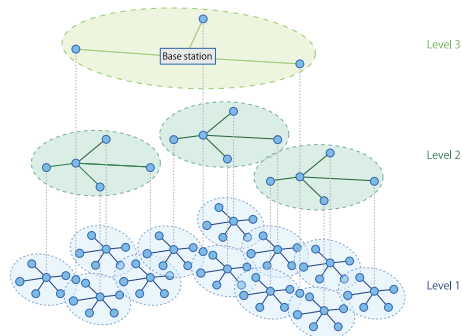


図1 階層的センサネットワーク
Fig. 1 An example of hierarchical sensor network.

ネットワークのトポロジは、基地局が無線通信の通信範囲や衝突、電力消費などを考慮してあらかじめ決定するものとする。提案手法は、基地局にネットワーク上の全センサノードからデータを収集する際に、センサデータの特徴を考慮して効率の良いデータ収集を実現するものであるため、センサノードの通信方式にかかわらず効果のある手法である。省電力なデータ収集方式としては、これまでに LEACH⁽⁶⁾ や HEED⁽¹⁴⁾ などさまざまなクラスタリングによるデータ収集手法が提案されているが、LEACH や HEED では、分散処理によってクラスタリングを確率的に行うことでネットワークトポロジを決定するため、木構造のトポロジに偏りが生じる可能性がある。本研究では特異値分解を階層的に用いてデータを集約するため、偏りが大きいとデータの圧縮効率が悪くなったり、誤差が大きくなったりする可能性がある。そこでたとえばランダム探索によるクラスタリングアルゴリズムである CLARANS⁽¹⁰⁾ を、無線通信の衝突と木構造の偏りの少なさを目的関数として階層的に適用するなどしてネットワークトポロジを生成し、得られた偏りの少ないネットワークトポロジをデータ収集に利用することを想定している。センサデータの類似性やセンサノードの空間的な配置を考慮したクラスタリングによってネットワークトポロジを決定すれば、さらに効率良くデータ収集が行える。

センサデータを集約していくうえでは、基地局はデータを収集する木構造のネットワークの構成を把握し、その情報をもとに各センサノードで発生したデータを復元する。センサノードは、基地局から指定された自身の親ノードと子ノードの情報を持っているものとする。実際に提案手法を運用するうえでは、トポロジが固定であると中継のセンサノードにお

ける無線の送受信の頻度が高く、末端のセンサノードよりも早く電池を消耗するため、役割を一定時間で交代することによってネットワークを長期間稼働できるようにする仕組みが必要であるが、本論文では扱わず、今後の課題とする。

3. 特異値分解を用いたセンサデータ系列の圧縮手法

3.1 センサネットワークにおけるデータ系列の圧縮

これまでに述べたように、多数のノードからなるセンサネットワークでは、隣接するセンサノードがクラスタを生成し、クラスタごとにデータを集約する階層的なネットワークトポロジが有効である。本論文では、センサデータの周期性と近接するセンサノード間のデータの類似性を利用してデータを圧縮して収集する手法を提案する。具体的には、特異値分解を用いてデータの集約を行う。特異値分解を用いると、センサデータ系列の代表的な変化を基底系列とし、新たなデータ系列を基底系列に対するわずかな数の重みの係数で近似でき、少ない誤差でデータを圧縮できる。フーリエ変換を使っても同じように三角関数に対する重みの係数でデータ系列を表現できるが、特異値分解を用いた方がセンサデータの特徴をとらえた系列を基底系列とすることができ、より効率の良いデータ収集が実現できる。

単体のセンサノードでは、一定時間ごとにデータ系列を区切り、複数個のデータ系列をまとめて特異値分解によって基底系列とそれに対する重みの係数値に分解する。周期的な繰返しのあるデータ系列では代表的な変化が基底の系列に現れ、効率良く近似できる。さらに本論文では、階層的に特異値分解を繰り返す、データを収集する手法を提案する。末端のセンサノードでは、発生したセンサデータを基底系列と係数値に分解し、親ノードに送信する。親ノードでは、受信した複数の子ノードからの基底系列をまとめ、特異値分解によって代表的な基底系列と係数値に分解する。この操作を階層的に繰り返すと、上位の階層では、センサノード間で共通の基底系列に集約される。階層的な特異値分解を用いた研究例としては、周期性のある時系列データから代表的なパターンを取り出す手法が提案されている⁽¹¹⁾。筆者らはこの研究例を参考に、センサネットワークにおけるセンサデータの集約に対して階層的に特異値分解を適用する手法を考案した。

本章で単体のセンサノードにおける特異値分解を利用したデータ圧縮手法について述べ、次章で階層的に特異値分解を適用して効率的にデータを収集する手法について述べる。

3.2 特異値分解

本節では、特異値分解を利用し、データを圧縮する提案手法の基本的なアイデアについて述べる。

本論文では、 n 個の要素からなるベクトルは $\mathbf{v} \equiv [v_1 v_2 \cdots v_n]^T$ と小文字のボールド体で表記し、 $m \times n$ の行列を \mathbf{A} と大文字のボールド体で表記する。行列 \mathbf{A} の列ベクトルは $\mathbf{a}_{(i)}$ 、行ベクトルは \mathbf{a}_j とする。つまり、 $\mathbf{A} \equiv [\mathbf{a}_{(1)} \mathbf{a}_{(2)} \cdots \mathbf{a}_{(n)}] \equiv [\mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_m]^T$ となる。

特異値分解は線形変換の一種であり、任意の $m \times n$ の行列 \mathbf{A} を、

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (1)$$

と分解できる。ここで \mathbf{A} のランクを r とすると、 \mathbf{U} は、 $m \times r$ のユニタリ行列、 \mathbf{V} は $n \times r$ のユニタリ行列、 $\mathbf{\Sigma}$ は $r \times r$ の対角行列である。 $\mathbf{\Sigma} \equiv \text{diag}[\sigma_1 \sigma_2 \cdots \sigma_r]$ は正の値 σ_i からなる対角行列であり、 \mathbf{A} の特異値を指す。対角成分 σ_i は降順に並べられている。 \mathbf{U} 、 \mathbf{V} はユニタリ行列であるため、各成分 $\mathbf{v}_{(i)}$ 、 $\mathbf{u}_{(i)}$ は単位ベクトルとなる。

特異値分解の適用例の 1 つとして、行列の近似があげられる。降順に並んだ特異値の上位 k 個のみで行列を表現すると、 \mathbf{A} と $\tilde{\mathbf{A}}$ を近似した行列 $\tilde{\mathbf{A}}$ の間の誤差 (Frobenius ノルム) が最小となる。

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T \quad (2)$$

ここで、 $\tilde{\mathbf{U}} := [\mathbf{u}_{(1)} \mathbf{u}_{(2)} \cdots \mathbf{u}_{(k)}]$ 、 $\tilde{\mathbf{V}} := [\mathbf{v}_{(1)} \mathbf{v}_{(2)} \cdots \mathbf{v}_{(k)}]$ 、 $\tilde{\mathbf{\Sigma}} = \text{diag}[\sigma_1 \sigma_2 \cdots \sigma_k]$ とする。 $\mathbf{A} - \tilde{\mathbf{A}}$ の Frobenius ノルムは以下の式で定義される。

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 = \sum_{i,j=1}^{m,n} |a_{ij} - \tilde{a}_{ij}|^2 = \sum_{i=k+1}^r \sigma_i^2. \quad (3)$$

式 (1) の変換において、 $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}$ とすれば、

$$\mathbf{A} = \mathbf{P}\mathbf{V}^T \quad (4)$$

と変形できる。この式は、 $\mathbf{v}_{(i)}$ を軸とした \mathbf{A} から \mathbf{P} への射影変換を意味し、 \mathbf{P} の各行は、変換後の座標系における \mathbf{A} の各行の座標を意味すると理解できる。つまり、 \mathbf{V} の上位 k 個のベクトルを用いて座標を変換することは、 \mathbf{A} の各点をより次元の低い k 次元の座標系へ射影することを意味する。同様にして、 $\tilde{\mathbf{P}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}$ とすると、

$$\tilde{\mathbf{A}} = \tilde{\mathbf{P}}\tilde{\mathbf{V}}^T \quad (5)$$

となる。

このように、 $\tilde{\mathbf{A}}$ は、 $\tilde{\mathbf{P}}$ と $\tilde{\mathbf{V}}$ を用いて近似できる。 $m > k$ かつ $n > k$ であるため、センサノードは $m \times n$ の \mathbf{A} の代わりに、より少ないデータ量の $m \times k$ の行列 $\tilde{\mathbf{P}}$ と $n \times k$ の行

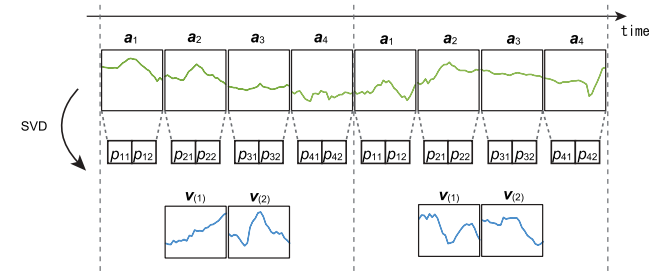


図 2 特異値分解によるデータの圧縮 ($c = 4$, $d = 0.5$)
Fig. 2 An example of data approximation using SVD ($c = 4$, $d = 0.5$).

列 $\tilde{\mathbf{V}}$ を送信するだけで、基地局ではセンサデータを復元できる。しかも、任意の k に対して Frobenius ノルムが最小となるため、その二乗誤差も最小となる。

3.3 データの圧縮

センサノードでは、連続したセンサデータ系列を長さ l のウィンドウに区切り、 c 個の部分系列を \mathbf{A} として保持する ($l > c$ とする)。センサデータで発生したデータ系列を $\{x_1, x_2, \dots, x_{cl}\}$ とすれば、 \mathbf{A} の i 番目の行 \mathbf{a}_i に部分データ系列 $\{x_{(i-1)l+1}, x_{(i-1)l+2}, \dots, x_{il}\}$ を保存することになる。この際に部分データ系列の平均と分散を算出して \bar{a}_i と s_i とし、正規化を行ったうえで保存する ($a_{ij} = (x_{(i-1)l+j} - \bar{a}_i) / s_i$)。 c 個のデータ系列が揃うと、式 (4) のように \mathbf{A} を特異値分解によって、 \mathbf{P} と \mathbf{V} に分解する。 \mathbf{V} から上位 k 個の $\mathbf{v}_{(i)}$ を基底となる系列として選択し、これに対する係数行列 $\tilde{\mathbf{P}}$ に \mathbf{A} を圧縮する。以上の処理を c 個のデータ系列が揃うたびにを行う。この単純な手法をバッチ処理手法と呼ぶ。

元のデータ系列 \mathbf{A} は $c \times l$ のデータ量があるのに対して、圧縮したデータサイズは $k \times l + k \times c$ となる。データ系列の個数 c に対する基底系列 k の割合を $d = k/c$ とすれば、元のデータ量に対する圧縮後のデータ量の割合 r は、

$$r = \frac{cd(l+c)}{cl} = \frac{d(l+c)}{l} \quad (6)$$

となる。

基地局では、式 (5) のようにセンサノードから受け取った $\tilde{\mathbf{V}}$ と $\tilde{\mathbf{P}}$ を掛け合わせて元のデータ系列 $\tilde{\mathbf{A}}$ を復元する。また、復元された $\tilde{\mathbf{A}}$ の元のデータとの誤差は、設定された圧縮率に対して最小に抑えられる。

図 2 に特異値分解を用いたデータ圧縮の例を示す。図 2 では、ある地点の温度センサの

データを例とし, 1 日ごとにウィンドウを区切っている. 4 つの部分系列 $\mathbf{a}_1, \mathbf{a}_4$ が, 特異値分解によって基底系列 $\mathbf{v}_{(1)}, \mathbf{v}_{(2)}$ と 8 つの係数値 p に圧縮されている. この例に限らず, 多くのセンサデータには周期性があり, 適切なウィンドウ幅 l を用いれば, d に小さな値を設定してデータ量を抑えても, 少ない誤差でデータを圧縮できる. 実際には, l や c はデータの周期とセンサノードのメモリサイズを考慮して決定する. 周期が前もって分からない場合には, 文献 11) などと提案されている自動的にデータの周期を解析する方法を用いれば適切な値を決定できる. d の値は, ネットワークの帯域やセンサノードのメモリサイズといった制約と, データの特性や誤差を許容する度合いによって決定する. 式 (3) より特異値を加算することで誤差が求められるため, これを利用して動的に k を決定するような手法も実現可能であるが, 本論文では固定の値とした.

3.4 オンライン処理手法

前節で述べたバッチ処理手法には, 以下のような改善の余地がある.

- 単純なバッチ処理手法では, センサデータがよく似たパターンを繰り返し, 基底系列 \mathbf{V} にほとんど変化がない場合でも, c 個のデータ系列ごとに $\tilde{\mathbf{V}}$ を無線通信で送信するため, 効率が悪い.
- c 個のデータが発生するごとに無線通信の処理を行うため, データ収集に遅延が生じる. このため, センサデータの傾向が変化するまで, 同じ $\tilde{\mathbf{V}}$ を繰り返し使用し, 傾向が変化したときにのみ $\tilde{\mathbf{V}}$ を送信しなすよう, バッチ処理手法を改良したオンライン処理手法を提案する. この手法では, センサノードは, データの傾向に変化がなければ, 大きなデータ量の $\tilde{\mathbf{V}}$ は低い頻度で送信し, 少ないデータ量の \mathbf{p}_i のみを送信すればよい. 大きく圧縮の効率を上げられる.

すでにある基底の系列 \mathbf{V} に対する重み \mathbf{P} は以下のようにして計算する. 式 (4) は, \mathbf{V} がユニタリ行列であるため,

$$\mathbf{P} = \mathbf{A}\mathbf{V} \quad (7)$$

と変形できる. この式から, $\tilde{\mathbf{P}}$ は

$$\tilde{\mathbf{P}} \approx \tilde{\mathbf{A}}\tilde{\mathbf{V}}. \quad (8)$$

によって近似できると考えられる. そこで, \mathbf{a}_i が更新されたときに, この \mathbf{a}_i を基底の系列の和で表現するための重みを,

$$\tilde{\mathbf{p}}_i = \mathbf{a}_i\tilde{\mathbf{V}} \quad (9)$$

と算出する. データを復元する際には, 式 (4) と同様にして,

```

1. Compute average ( $\bar{a}_i$ ) and variance ( $s_i$ ) of  $\{x_{t+1}, \dots, x_{t+l}\}$ ;
2. for  $j = 1$  to  $l$ 
3.    $a_{ij} = (x_{t+j} - \bar{a}_i)/s_i$ ; // Normalization
4.  $\tilde{\mathbf{p}}_i = \mathbf{a}_i\tilde{\mathbf{V}}$ ;
5.  $\tilde{\mathbf{a}}_i = \tilde{\mathbf{p}}_i\tilde{\mathbf{V}}^T$ ;
6.  $e = \|\mathbf{A} - \tilde{\mathbf{A}}\|^2$ ;
7. if  $e < \epsilon$  then
8.   Send  $\tilde{\mathbf{p}}_i, \bar{a}_i$ , and  $s_i$ ;
9. else
10.  Compute SVD of  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}'^T$ ;
11.   $\mathbf{P}' = \mathbf{U}\mathbf{\Sigma}$ ;
12.   $\tilde{\mathbf{A}}' = \tilde{\mathbf{P}}' \times \tilde{\mathbf{V}}'^T$ ;
13.   $e' = \|\mathbf{A} - \tilde{\mathbf{A}}'\|^2$ ;
14.  if  $e' > e$  then
15.    Send  $\tilde{\mathbf{p}}_i, \bar{a}_i$ , and  $s_i$ ;
16.  else
17.     $\tilde{\mathbf{V}} = \tilde{\mathbf{V}}'$ ; // Update base sequences.
18.    Send  $\tilde{\mathbf{p}}'_i, \bar{a}_i, s_i$ , and  $\tilde{\mathbf{V}}$ ;
19.  end if
20. end if
21.  $i++$ ;
22. if  $i == c$  then  $i = 1$ ;

```

i は, \mathbf{A} にデータ系列からデータを入れる際のインデックス.

$\mathbf{P}', \mathbf{V}', \mathbf{A}', e'$ は一時変数.

x_i がこれまでに処理した最新のセンサ値.

図 3 オンライン処理アルゴリズム
Fig. 3 Online processing algorithm.

$$\tilde{\mathbf{a}}_i = \tilde{\mathbf{p}}_i\tilde{\mathbf{V}}^T \quad (10)$$

と計算する. 新たなデータ系列 \mathbf{a}_i がこれまでと同じ傾向であればこのような方法で算出した $\tilde{\mathbf{p}}_i$ を用いて少ない誤差で近似できるが, データ系列の傾向に大きな変化があった場合には, 上式のような算出方法では誤差が大きくなるため, 基底系列を更新する必要がある.

図 3 にオンライン処理アルゴリズムを示す. センサノードで長さ l の新しいデータ系列が発生すると, 正規化を行ったデータを \mathbf{a}_i に保持する (図 3 の 1-3 行目). ここで i は \mathbf{A} の各行に順にデータ系列を代入するためのインデックスである. 次に式 (9) を用いて基底の

系列に対する重み \tilde{p}_i を計算する (4 行目). 算出した \tilde{p}_i を用いて \tilde{A} をローカルに算出し, A との誤差 e を求める (5-6 行目). 誤差 e が閾値 ϵ (たとえば評価実験では 0.1 から 0.2 とした) 以下であれば, データ系列の傾向はこれまでと変化していないと見なし, \tilde{p}_i , \tilde{a}_i , s_i を送信する (7-8 行目). 閾値 ϵ を超えた場合は, データの系列の傾向に変化があったと見なし, 基底系列を更新する (9-20 行目). ただし, この処理を行う際には実際に誤差が少なくなると見込める場合にのみ更新を行う. 更新予定の基底系列を特異値分解によって求めて V' とし, これを用いて更新した場合の誤差 e' を算出して (10-13 行目), 実際に誤差が小さくなっているかを確認する. 誤差の減少が見込めないときには, 基底系列 V の更新は行わず, 誤差 e が ϵ 以下であった場合と同様に, \tilde{p}_i , \tilde{a}_i , s_i のみを送信する (14-15 行目). 誤差が減少した場合は, \tilde{V} を \tilde{V}' で上書きし, 新たな基底系列 \tilde{V} も同時に基地局へ送信する (16-19 行目). 基底系列の更新時の特異値分解の計算, および最初に c 個のデータ系列が発生するまではバッチ処理と同じアルゴリズムを用いる. また, 図 3 では行列の演算の際に, 行列全体を計算しているように記述されているが, 実際には, 新たなデータ系列によって変更された部分のみを計算している. なお, ϵ は, アプリケーションで求めるデータの精度によって決定するものとする.

この手法では, 誤差の大きさを検討する際に, 新しく発生したデータ系列のみを用いて復元したデータの誤差を計算するのではなく, 最近の c 個のデータ系列を用いて誤差を計算している. 一時的にデータ系列の傾向が変化するたびに新たな \tilde{V} を送信すると効率が悪いので, 最近の c 個で誤差が大きくなることを確認してから基底系列を更新する. 一時的に傾向の変化があった際に誤差が増加する可能性はあるが, その後, 元の傾向に戻る場合がある可能性もあることを考慮してこのような手法としている. また, センサノードでは, 長さ l のデータが発生するとすぐに, \tilde{V} を用いて p_i を計算してデータを送信できるため, データ収集の遅延低減にもつながる.

図 4 に実際の基底系列更新の例を示す. $l = 24$, $c = 16$, $d = 0.25$, $\epsilon = 0.2$ とした. 一番上のグラフが元のデータの系列であり, その左下のグラフが時刻 168 までの基底系列で復元した結果, その右下のグラフが時刻 168 以降の更新した基底系列で復元した結果である. 最下段には, 時刻 168 までの基底系列 ($v_{old(1)}$ から $v_{old(4)}$) と時刻 168 以降の基底系列 ($v_{new(1)}$ から $v_{new(4)}$) を示し, 重みの大きい主な基底系列を復元した系列に重ねて記載している. この例では, 時刻 168 から 172 で発生した 1 日の間に 2 回の温度の低下のピークがあるという新たな傾向を持つデータ系列が今までの基底系列では十分に復元できず, 累積の誤差が閾値を超えたため新たな基底系列に更新している. 更新後の基底系列には $v_{new(3)}$

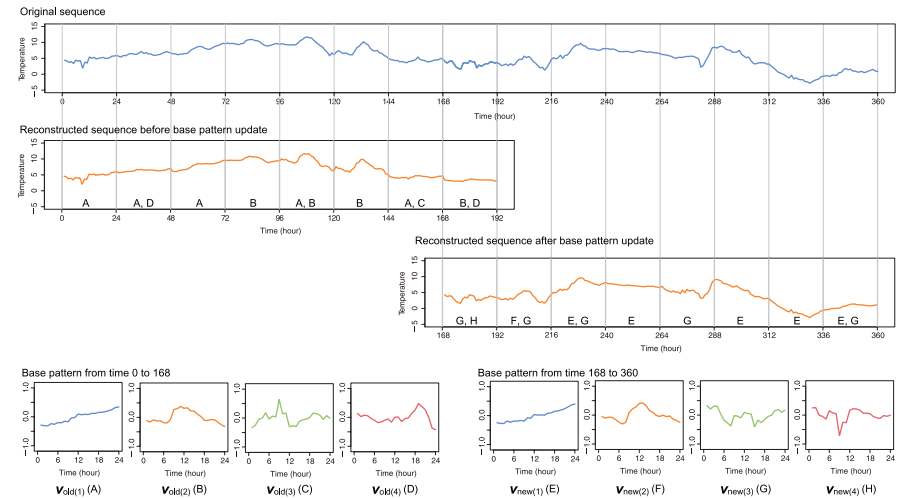


図 4 基底系列更新の例 (AMEDAS のある地点の温度計のデータを利用)

Fig. 4 An example of local pattern updating (the original sequence is outdoor temperature data).

のように時刻 168 から 172 のデータ系列の特徴をよく表す系列が含まれ, この区間の更新前と更新後の基底系列で復元したデータを比較すると, 更新前の復元系列ではほぼ平坦なデータ系列になり気温の誤差の二乗平均が 52.2 となるのに対して, 更新後の復元系列では 2 回の温度低下のピークがよく復元されるようになり気温の誤差の二乗平均も 0.5 に減少している. しかも, 更新後の基底系列を用いてその後のデータ系列もよく近似できている.

4. 階層的特異値分解によるセンサデータ系列の圧縮

前章で述べた単体のセンサノードでのデータ圧縮手法を階層的に適用し, センサネットワーク全体からデータを集約する手法について述べる.

4.1 階層的なデータの集約

図 5 に階層的に特異値分解を適用したデータの集約手法のイメージ図を示す. 基本的な処理は単体のセンサノードでデータを集約する場合と同様である. 以降では, h 階層目のデータ系列を $A^{(h)}$, 基底系列を $V^{(h)}$ などとし, センサノード i を $N_i^{(h)}$ と表記する. また, 特にセンサノード i を区別して各変数を記述するときには, h 階層目の i 番目のセンサノードの基底系列を $V_i^{(h)}$, 同じセンサノードの 1 番目のデータ系列を $a_{i,1}^{(h)}$ などと表記する. こ

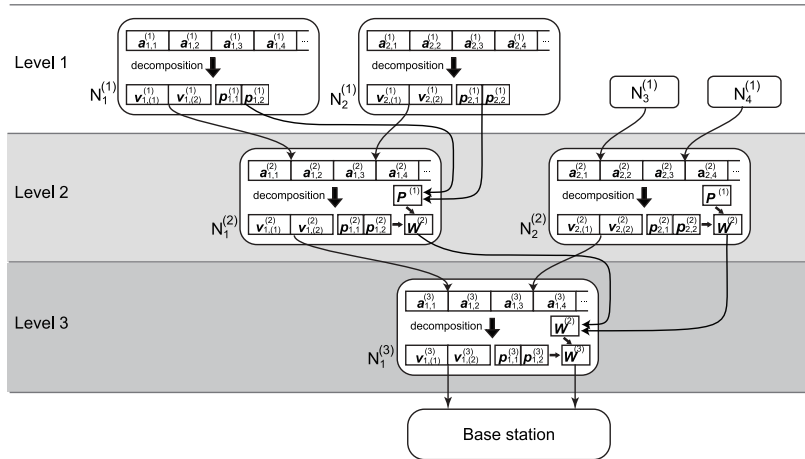


図5 階層的特異値分解によるデータの圧縮 (2分木のトポロジ, $c = 4, d = 0.5, h = 3$)
 Fig. 5 Data compression by hierarchical SVD (binary tree topology, $c = 4, d = 0.5, h = 3$).

ここでは最下位層のセンサノードで発生したデータを集約する手法について述べるが、途中の階層のノードで発生したセンサデータについても各階層で並行して処理を行ってデータを収集する。たとえば2階層目のセンサノードでは、1階層目で発生したデータを中継する処理だけでなく、自ノードで発生したデータを圧縮して送信する処理も行う。

本節では、バッチ処理手法を階層的に適用する手法について述べる。1階層目で発生したセンサデータは3.2節で述べた方法によって基底系列 $\tilde{\mathbf{V}}^{(1)}$ とこれに対する重みの係数値行列 $\tilde{\mathbf{P}}^{(1)}$ に分解され、親ノードである2階層目のノードへ送信される。 $\mathbf{A}^{(1)}$ の平均と分散は、そのまま基地局まで他のデータとともに転送される。2階層目のノードでは、受け取った基底系列を $\mathbf{A}^{(2)}$ に保存し、子ノード1から n の基底系列をまとめて特異値分解する。また、1階層目ですでに正規化は行っているため、2階層目以降では正規化は行わない。

$$\mathbf{A}^{(2)} = \begin{bmatrix} \tilde{\mathbf{V}}_1^{(1)} \\ \tilde{\mathbf{V}}_2^{(1)} \\ \vdots \\ \tilde{\mathbf{V}}_n^{(1)} \end{bmatrix} = \mathbf{P}^{(2)} \times \mathbf{V}^{(2)\top}. \quad (11)$$

得られた $\tilde{\mathbf{V}}^{(2)}$ を自身の親ノードである3階層目のセンサノードへ送信する。子ノードは

近接した場所に設置され、基底系列にも類似性がある可能性が高く、少ない誤差で圧縮できると予想される。

次に重みの係数値行列 $\tilde{\mathbf{P}}^{(1)}$ の転送について述べる。2階層目では、 i 番目の子ノードのデータ系列 $\tilde{\mathbf{A}}_i^{(1)}$ は、各子ノードから k 行ずつのデータを受け取るため、

$$\tilde{\mathbf{A}}_i^{(1)} = \tilde{\mathbf{P}}_i^{(1)} \times \tilde{\mathbf{V}}^{(1)\top} = \tilde{\mathbf{P}}_i^{(1)} \times \left[\tilde{\mathbf{a}}_{(i-1)k+1}^{(2)}, \tilde{\mathbf{a}}_{(i-1)k+2}^{(2)}, \dots, \tilde{\mathbf{a}}_{ik}^{(2)} \right]^\top \quad (12)$$

と復元できる。つまり式(11)より、

$$\tilde{\mathbf{A}}_i^{(1)} = \tilde{\mathbf{P}}_i^{(1)} \times \left[\tilde{\mathbf{p}}_{(i-1)k+1}^{(2)}, \tilde{\mathbf{p}}_{(i-1)k+2}^{(2)}, \dots, \tilde{\mathbf{p}}_{ik}^{(2)} \right]^\top \times \tilde{\mathbf{V}}^{(2)\top} \quad (13)$$

となり、 $\tilde{\mathbf{P}}^{(1)}$ 、 $\tilde{\mathbf{P}}^{(2)}$ 、 $\tilde{\mathbf{V}}^{(2)}$ を掛け合わせることで、データ系列 $\tilde{\mathbf{A}}^{(1)}$ が復元できる。つまり、1階層目で発生した重みの係数値行列 $\tilde{\mathbf{P}}^{(1)}$ は、2階層目において係数行列 $\tilde{\mathbf{W}}^{(2)}$ として、 $\tilde{\mathbf{P}}^{(2)}$ と掛け合わせたうえで次の階層に転送すればよい。これより、各階層で、センサノード i を子孫に持つ $h-1$ 階層のセンサノードが h 階層の中間ノードの j 番目の子ノードであるとき、 $\tilde{\mathbf{W}}_i^{(h)}$ を

$$\tilde{\mathbf{W}}_i^{(h)} = \tilde{\mathbf{W}}_i^{(h-1)} \times \begin{bmatrix} \tilde{\mathbf{P}}_{(j-1)k+1}^{(h)} \\ \tilde{\mathbf{P}}_{(j-1)k+2}^{(h)} \\ \vdots \\ \tilde{\mathbf{P}}_{jk}^{(h)} \end{bmatrix} \quad (14)$$

と計算すれば、 $\tilde{\mathbf{W}}_i^{(h)}$ を用いて

$$\tilde{\mathbf{A}}_i^{(1)} = \tilde{\mathbf{W}}_i^{(h)} \times \tilde{\mathbf{V}}^{(h)\top} \quad (15)$$

とデータ系列が復元できる。

これを階層的に繰り返す、最終的に基地局には、最上位の階層で計算された基底系列 $\tilde{\mathbf{V}}^{(h_{max})}$ 、各センサノードで発生したデータ系列の $\tilde{\mathbf{V}}^{(h_{max})}$ に対する重みの係数値行列 $\tilde{\mathbf{W}}^{(h_{max})}$ と、1階層目で発生した平均と分散が収集される。基地局は式(15)のように $\tilde{\mathbf{W}}^{(h_{max})}$ と $\tilde{\mathbf{V}}^{(h_{max})}$ を掛け合わせてデータ系列 $\tilde{\mathbf{A}}^{(1)}$ を復元し、受信した平均と分散を用いて1階層目で発生した元のデータ系列を復元できる。なお、ここでは説明の都合上、全階層の全センサノードで同じ k を用いる場合について説明したが、階層やセンサノードごとに異なる k を用いてもかまわない。

以上が階層的にバッチ処理を繰り返す手法である。階層的にデータ系列を処理しても、基

基底系列に変化がない場合でも繰り返し基底系列を送ってしまうという問題点があるため、次節で述べるように、データ系列の傾向に変化があったときのみ基底系列を送信するオンライン処理による手法を提案する。

4.2 階層的オンライン処理手法

図6にオンライン処理手法によるデータ収集手法のアルゴリズムを示す。バッチ処理手法と同様にしてオンライン処理によるデータ圧縮を階層的に適用するアルゴリズムとなっている。図6に示したのは、中間の階層 h のあるノードにおいて、 $h-1$ 階層の i 番目の子ノードから更新された基底系列 $\tilde{\mathbf{V}}_i^{(h-1)}$ と、 $h-1$ 階層で計算された係数値行列 $\mathbf{w}^{(h-1)}$ を受信した際のデータ処理である。中間ノードでは、受信した $\tilde{\mathbf{V}}_i^{(h-1)}$ を $\mathbf{A}^{(h)}$ に保存し(1-2行目)、現在の基底 $\mathbf{V}^{(h)}$ を用いて近似した際の誤差 e を計算する(3-5行目)。誤差が閾値 ϵ よりも大きければ、傾向が変化すると判断し、再度特異値分解を行い基底系列の更新を行う(6-15行目)。この際、図3と同様に新たな基底系列によって誤差が小さくなることを確認し(7-11行目)、誤差が小さくなる場合のみ、基底系列を $\tilde{\mathbf{V}}^{(h)}$ を更新して自身の親

```

1. for  $j = 1$  to  $k$ 
2.    $\mathbf{a}^{(h)}_{(i-1)k+j} = \tilde{\mathbf{v}}_{i,j}^{(h-1)}$ ;
3.    $\tilde{\mathbf{P}}^{(h)} = \mathbf{A}^{(h)} \times \tilde{\mathbf{V}}^{(h)}$ ;
4.    $\tilde{\mathbf{A}}^{(h)} = \tilde{\mathbf{P}}^{(h)} \times \tilde{\mathbf{V}}^{(h)\top}$ ;
5.    $e = \|\mathbf{A}^{(h)} - \tilde{\mathbf{A}}^{(h)}\|^2$ ;
6.   if  $e > \epsilon$  then
7.     Compute SVD of  $\mathbf{A}^{(h)} = \mathbf{U}^{(h)} \mathbf{\Sigma}^{(h)} \mathbf{V}^{(h)\top}$ ;
8.      $\tilde{\mathbf{P}}'^{(h)} = \tilde{\mathbf{U}}^{(h)} \tilde{\mathbf{\Sigma}}^{(h)}$ ;
9.      $\tilde{\mathbf{A}}'^{(h)} = \tilde{\mathbf{P}}'^{(h)} \times \tilde{\mathbf{V}}'^{(h)\top}$ ;
10.     $e' = \|\mathbf{A}^{(h)} - \tilde{\mathbf{A}}'^{(h)}\|^2$ ;
11.    if  $e' < e$  then
12.       $\tilde{\mathbf{V}}^{(h)} = \tilde{\mathbf{V}}'^{(h)}$ ,  $\tilde{\mathbf{P}}^{(h)} = \tilde{\mathbf{P}}'^{(h)}$ ; // Update base sequences.
13.      Send  $\tilde{\mathbf{V}}^{(h)}$ ;
14.    end if
15.  end if
16. Send  $\mathbf{W}^{(h)}$  computed by Equation(14)

```

$\tilde{\mathbf{P}}^{(h)}$, $\mathbf{V}^{(h)}$, $\mathbf{A}'^{(h)}$, e' は一時変数。

図6 階層的オンライン処理アルゴリズム

Fig.6 Hierarchical online processing algorithm.

ノードへ送信する(11-14行目)。最後に式(14)を用いて、 $\tilde{\mathbf{P}}^{(h)}$ と $\mathbf{W}^{(h-1)}$ から $\mathbf{W}^{(h)}$ を算出して上位階層へ転送する(16行目)。基底系列に更新がなく、 \mathbf{W} のみを受信したときには、最新の係数行列 \mathbf{P} を用いて16行目の処理のみを行う。また、図6の行列演算においても、実際には行列全体ではなく更新部分のみを計算する。

5. 評価実験

シミュレータを用いた評価実験により提案手法の有効性を検証した。シミュレーションでは、完全4分木のネットワークを想定し、末端の各センサノードで発生したセンサデータを1台の基地局に収集する際のデータの圧縮率と誤差の割合を評価した。実験で用いたデータセットの詳細を表1に示す。AMEDASの2007年7月から2008年6月の気象データから欠損の少ない895点の温度のデータと、38点で屋内の気温を約10日間計測したデータを用いた。両データとも観測地点のIDの順に繰り返しシミュレータ上のセンサノードにデータを割り当てた。シミュレータ上のセンサノードよりもデータセットのノード数が少ない場合には、1つの観測地点のデータがシミュレータ上の複数のセンサノードに割り当てられる。実験では $c = 16$, $d = 0.25$ とした。

圧縮率に関する評価は、階層的なデータ集約においてデータの圧縮を行っているのは基底系列 \mathbf{V} に関してのみであり、係数行列 \mathbf{W} はバッチ処理手法とオンライン処理手法で同じように上位階層へ転送しているため、基底系列のみを対象とした。ここで圧縮率は、全センサノードで発生したセンサデータ系列のデータ量に対する、基地局が直接受け取った基底系列のデータ量の割合である。

図7にAMEDASのデータを用いた場合の実験結果を、図8に屋内の気温データを用いた場合の実験結果を示す。実験では、階層数 $h = 1 \dots 6$ の4分木のトポロジを用意し、各階層において 4^{6-h} 回実験を行いその平均を求めた。図7と図8の(a)では、横軸はセンサノード数を対数で、縦軸は圧縮率(百分率)を対数で表示し、(b)では横軸はセンサノード

表1 実験データ
Table 1 Data set.

データの種類	AMEDAS	屋内の気温
ノード数	895	38
データ長	87600 (10分間隔, 1年間)	30000 (30秒間隔)
データの周期	144 (1日)	2880 (1日)
ウィンドウサイズ l	72	240

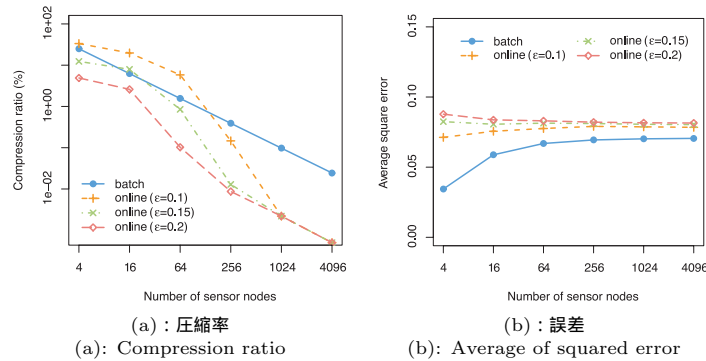


図 7 実験結果 (AMEDAS)
Fig. 7 Simulation result (AMEDAS).

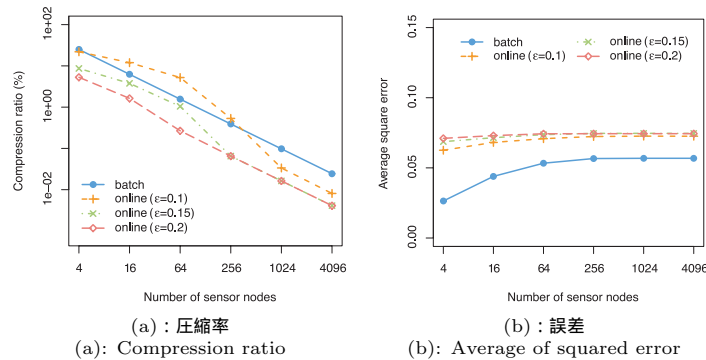


図 8 実験結果 (屋内の気温)
Fig. 8 Simulation result (indoor temperature).

数を対数で、縦軸は正規化したデータにおける元のデータ系列と復元したデータ系列の間の二乗誤差の平均を表示している。どちらのデータの場合も、全体的にバッチ処理よりもオンライン処理による手法の方が誤差は多いものの、圧縮の効率が良いことが確認できる。 ϵ については、値を小さくすると低い階層で誤差が減少するが、データ量は全体的に増加する。これより、提案手法を運用する際には、 ϵ の値を小さくしても誤算減少の度合いには限界があることを考慮して、 ϵ の値を決定する必要があることが分かった。また、圧縮率は階層数が増えるにつれて向上するが、特にオンライン処理手法の場合には誤差は増加していない。

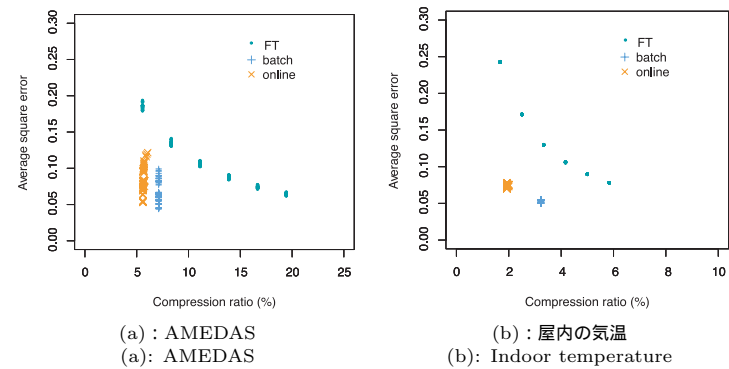


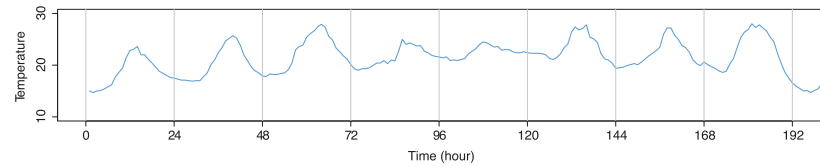
図 9 フーリエ変換による圧縮と提案手法の比較結果 (3 階層, $\epsilon = 0.2$)
Fig. 9 Comparison between Fourier transformation and the proposed method (3 levels, $\epsilon = 0.2$).

このことから提案手法は、多数のノードからなるセンサネットワークにおいて、センサノードの数が増加したとしても効率良くデータを収集できる手法であることが明らかになった。

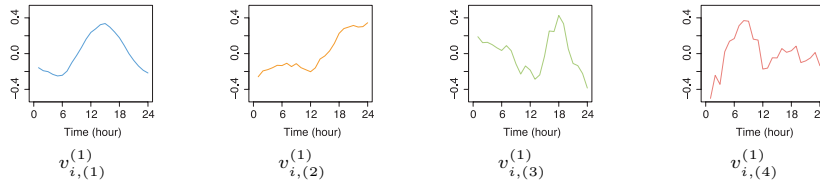
次に、3階層のトポロジで、提案手法とフーリエ変換を比較した結果を図9に示す。このグラフは、提案手法では64回の試行について、フーリエ変換では同じデータセットを用いて変換後の係数値を周波数が低いものから数個をデータの近似に用いた場合の圧縮率と誤差の関係を図示している。提案手法で基底系列に対する重みの値は圧縮せずにすべてのデータを基地局に送信しているのと同じように、フーリエ変換後の係数値をそのまま基地局に送信するものとした。圧縮率は、提案手法では基地局が受信した基底系列 V と係数行列 W の合計の全センサデータ量に対する割合、フーリエ変換では係数値のデータ量の全センサデータ量に対する割合としている。グラフの横軸が圧縮率、縦軸が平均の二乗誤差である。実験の結果、提案手法はフーリエ変換よりも、同等の圧縮率であればより少ない誤差で、同等の誤差であるならばより少ないデータ量でセンサデータを収集できることが明らかになった。

また、6階層のトポロジでシミュレーションを行った際の実際のデータ例を図10、図11に示す。この例では、AMEDASのデータは1時間に1点にデータを削減したものをを用い、 $l = 24$ とし、屋内の気温のデータでは $l = 120$ とした。いずれのグラフも横軸は時刻、縦軸はセンサデータあるいは、特異値分解によって生成された基底系列の値である。(a)がセンサノード i で発生した元のデータ系列、(b)が(a)のデータ系列を特異値分解して得られた基底の系列である。 $c = 16$, $d = 0.25$ のとき $k = cd = 4$ となるため、この4つの系列が

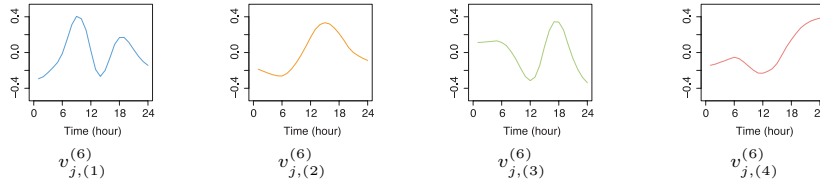
91 階層的センサネットワークのための効率的なデータ収集手法



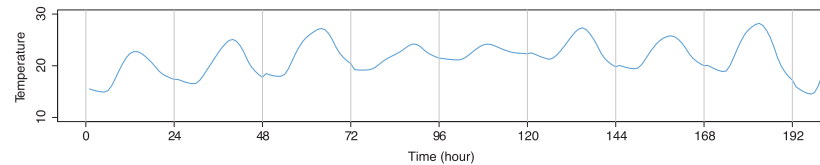
(a): センサノード N_i で発生したデータ系列
(a): The original data sequence of sensor node N_i at the first level



(b): (a) を特異値分解して得られた基底系列
(b): Local patterns computed using SVD of (a)



(c): 6 階層目のノード N_j における基底系列 (ノード N_j は最下層目のノード N_i の祖先)
(c): Local patterns on N_j at sixth level (N_j is an ancestor of N_i)

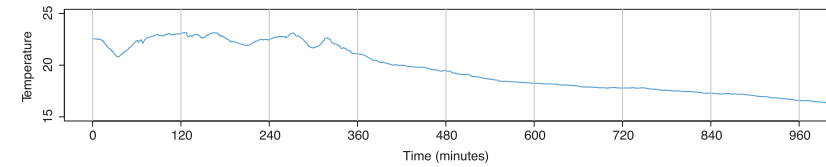


(d): ノード N_i の復元したデータ系列
(d): Reconstructed data sequence of N_i

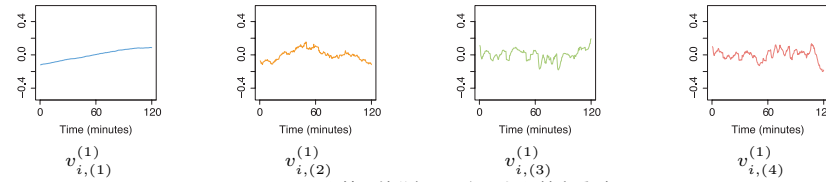
図 10 データの例 (AMEDAS)

Fig. 10 Example of data processing (AMEDAS).

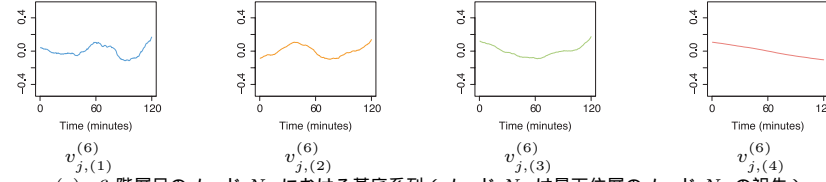
基地局に送信される。(c) は 6 階層目に集約された基底系列である。複数のセンサノードのデータ系列から平均的な系列が 6 階層目の基底系列として現れるため、各ノードからの基底系列が平滑化されたような波形となっている。(d) が (c) を用いて復元されたノード i のデータ系列である。(c) が平滑化されたような波形となっているため、データの細かな変化



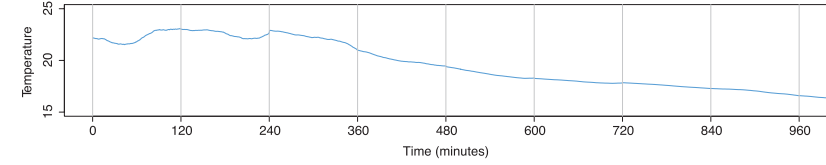
(a): センサノード N_i で発生したデータ系列
(a): The original data sequence of sensor node N_i at the first level



(b): (a) を特異値分解して得られた基底系列
(b): Local patterns computed using SVD of (a)



(c): 6 階層目のノード N_j における基底系列 (ノード N_j は最下層目のノード N_i の祖先)
(c): Local patterns on N_j at sixth level (N_j is an ancestor of N_i)



(d): ノード N_i の復元したデータ系列
(d): Reconstructed data sequence of N_i

図 11 データの例 (屋内の気温)

Fig. 11 Example of data processing (indoor temperature).

は失われている部分もあるが、元のセンサデータを十分に復元できた。特に図 11 では、センサデータの変化の周期とウィンドウ長 l が近い AMEDAS データの場合と異なり、ウィンドウの中にも上下の値の変化がある区間や単調な変化しかない区間が混ざっているような場合でもセンサデータをよく復元できることが確認できた。

6. ま と め

本論文では、センサネットワークにおける特異値分解を用いた効率的なデータ収集手法について述べた。提案手法は、階層的な木構造のネットワークポロジのセンサネットワークにおいて、特異値分解を階層的に適用してデータを圧縮し、少ないデータ量で効率的にセンサデータを収集する。シミュレーションによる評価実験の結果、提案手法でデータを収集すると、センサノードの数が増加して階層が増えてもほとんど変わらない誤差でデータを収集でき、しかもそのデータ量も低く抑えられることを明らかにした。さらにフーリエ変換による圧縮と提案手法を比較し、より効率的にデータを収集できることを示した。

このように、提案手法によって多数のセンサノードを用いたアプリケーションを構築する際に、容易にデータを収集できるセンサネットワークが構築できるようになるため、提案手法によってセンサネットワークの適用範囲を広げることができる。

参 考 文 献

- 1) Beigl, M. and Gellersen, H.: Smart-Its: An Embedded Platform for Smart Objects, *Smart Objects Conference (sOc) 2003* (2003).
- 2) Deligiannakis, A., Kotidis, Y. and Roussopoulos, N.: Hierarchical In-Network Data Aggregation with Quality, *Proc. International Conference on Extending Database Technology*, pp.577–587 (2004).
- 3) Deligiannakis, A., Kotidis, Y. and Roussopoulos, N.: Dissemination of compressed historical information in sensor networks, *The VLDB Journal*, Vol.16, No.4, pp.439–461 (2007).
- 4) Fan, K.-W., Liu, S. and Sinha, P.: Scalable Data Aggregation for Dynamic Events in Sensor Networks, *Proc. 4th International Conference on Embedded Networked Sensor Systems (Sensys 2006)*, pp.181–194 (2006).
- 5) Gupta, H., Navda, V., Das, S.R. and Chowdhary, V.: Efficient gathering of correlated data in sensor networks, *Proc. International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2005*, pp.402–413 (2005).
- 6) Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *Proc. 33rd Hawaii International Conference on System Sciences*, Vol.8, p.8020 (2000).
- 7) Madeen, S., Franklin, M.J., Hellerstein, J.M. and Hong, W.: TAG: A Tiny AGgregation Service for Ad-hoc Sensor Networks, *Proc. Symposium on Operating Systems Design and Implementation*, pp.131–146 (2002).
- 8) Madden, S., Szewczyk, R., Franklin, M.J. and Culler, D.: Supporting Aggregate

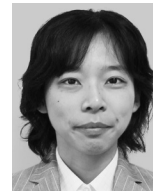
Queries Over Ad-hoc Wireless Sensor Networks, *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp.49–58 (2002).

- 9) Munguia, E., Stephen, T. and Larson I.K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors, *Proc. Pervasive 2004*, pp.158–175 (2004).
- 10) Ng, R.T. and Han, J.: Efficient and Effective Clustering Methods for Spatial Data Mining, *Proc. VLDB'94*, pp.144–155 (1994).
- 11) Papadimitriou, S. and Yu, P.: Optimal multi-scale patterns in time series streams, *Proc. 2006 ACM SIGMOD international conference on Management of data*, pp.647–658 (2006).
- 12) Rajagopalan, R. and Varshney, P.K.: Data-aggregation Techniques in Sensor Networks: A Survey, *IEEE Communications Surveys & Tutorials*, Vol.8, Issue 4, pp.48–63 (2006).
- 13) Wilson, D. and Atkeson, C.: Simultaneous Tracking & Activity Recognition (STAR) Using Many Anonymous, Binary Sensors, *Proc. Pervasive 2005*, pp.62–79 (2005).
- 14) Younis, O. and Fahmy, S.: HEED: A hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks, *IEEE Trans. Mobile Computing*, Vol.3, No.4, pp.366–379 (2004).

(平成 22 年 6 月 20 日受付)

(平成 22 年 10 月 13 日採録)

(担当編集委員 上善 恒雄)



岸野 泰恵 (正会員)

平成 14 年大阪大学工学部卒業。平成 16 年同大学院情報科学研究科博士前期課程修了。平成 19 年同研究科博士後期課程修了、日本電信電話株式会社入社。博士 (情報科学)。ユビキタスコンピューティング、センサネットワークに関する研究に従事。



櫻井 保志 (正会員)

平成 3 年同志社大学工学部電気工学科卒業。同年日本電信電話株式会社入社。平成 11 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。博士 (工学)。平成 16~17 年カーネギーメロン大学客員研究員。本会平成 18 年度長尾真記念特別賞, 平成 16 年度および平成 19 年度論文賞, 電子情報通信学会平成 19 年度論文賞, 日本データベース学会上林奨励賞, KDD 2008 best paper award runner-up 等受賞。索引技術, データストリーム処理, センサデータ処理技術の研究に従事。ACM, 電子情報通信学会, 日本データベース学会各会員。



亀井 剛次

平成 9 年京都大学大学院工学研究科修士課程修了。同年日本電信電話 (株) 入社。NTT コミュニケーション科学基礎研究所にて情報共有に基づくコミュニティ形成支援, センサネットワークによる実世界状況理解のための知識構築の研究に従事。平成 21 年より ATR 知能ロボティクス研究所にてロボットのいる実世界環境でのインタラクションデザインの研究に従事。電子情報通信学会および人工知能学会各会員。



前川 卓也 (正会員)

平成 15 年大阪大学工学部情報システム工学科卒業。平成 18 年同大学院情報科学研究科博士後期課程修了。同年日本電信電話株式会社入社。博士 (情報科学)。ユビキタスコンピューティング, Web 情報処理の研究に興味を持つ。



柳沢 豊 (正会員)

平成 10 年大阪大学大学院工学研究科博士後期課程修了。同年 NTT 基礎研究所入所。平成 12 年より NTT コミュニケーション科学基礎研究所配属。主任研究員。プログラミング言語, センサネットワーク, データ工学の研究に従事。博士 (工学)。IEEE 会員。



岡留 剛 (正会員)

昭和 63 年東京大学大学院情報科学科博士課程修了。同年 NTT 基礎研究所入所。以来, 人間の情報出力過程と調音運動の解明・形式言語の学習・センサネットワークを利用したコンテンツの自動生成の研究に従事。NTT コミュニケーション科学基礎研究所主幹研究員 (グループリーダー) を経て 2009 年 4 月より関西学院大学理工学部教授。博士 (理学)。センサネットワーク, 環境メディアデザイン, ユビキタスコンピューティングに興味を持つ。ACM, 日本認知科学会各会員。