ブラウジングのための2次元情報分布空間作成の高速化と一覧性の改善について

柿元俊博[†] 上原祐介[††] 上林彌彦[†††]

現在インターネット上の情報は年に4〜8倍という速度で増加しつつあり，このような大量な情報の中から利用者の要求を満足する情報を効率良く発見することがますます重要となってきている．しかしながら，テキスト以外のマルチメディア情報を扱う場合，検索キーによる方法では不十分でブラウジングとの併用が不可欠である．本稿では自己組織化マップを高速に計算し，利用者に分かりやすい形で提示できるように改良を行ったので報告する．利用者に情報全体の概観を提供することを第1の目的とするため，自己組織化マップの計算に精度が要求されないことを利用して，学習最適位置の計算の削減とサンプル値の利用で速度の向上を図った．10,000件のデータを対象とした例では計算速度を500倍くらいにできることが示された．この結果，通常の能力のパーソナルコンピュータで10,000件程度のデータを実用的な時間で扱うことが可能である．

Improvement of Generation Speed and Browsability of Browsing Space

Toshihiro Kakimoto,[†] Yusuke Uehara[††] and Yahiko Kambayashi[†††]

It is important to find out the information required by a user effectively from such a lot of information that is increasing from 4 to 8 times speed on the Internet. In the case of retrieving multimedia data, it, however, is insufficient to retrive its data by search key, and they are necessary both to retrieve it by search key and to browse it. This paper reports the improvement of speed to compute the self-organizing map (SOM) and the browsability of 3D browsing space. In order that the overviewing all the information is the most important, it is not so important to compute SOM precicely, thereby we realized the speed up of computing SOM by the deduction of amount of computing the best position of learning and the utilization of sampling data. It is shown that SOM of 10,000 number of data is computed at 500 times speed than SOM_PAK. This result shows to be able to compute SOM of 10,000 number of data using the personal computer of entry class.

1. はじめに

最近，インターネットを代表とするネットワーク上のビジネスが本格化し，ネットワーク上を流れる情報は年4〜8倍という速度で増加しつつあり，それらを蓄積したり検索したりする技術がますます重要になってきている．しかし，いまだに情報検索の中心はキーワードなどの検索キーをベースにした検索システムであり，その利用方法も大きく変わっていない．また，マルチメディア情報も増加しているがテキスト以外のメディアに関しては一般的には精度が十分でない．ここでは，情報検索をブラウジングの観点から見直し，実用的な手法とするための技術について報告する．

情報を探す方法としては検索キーを指定して，その検索キーに適合する情報に絞る方法と，情報を代表するような表現をある基準により配置した空間をブラウジングして探す方法がある．前者には，キーワードを指定して，そのキーワードを含むテキスト情報を抽出するキーワード検索があり，後者には，図書に利用される十進分類や書棚による分類などの階層的な分類に従ってブラウジングする方法がある．これまでの情報検索の研究の中でブラウジングを取り上げたものとして，Xeroxの Scatter/Gather システム[1),2)]やMITの

† 富士通研究所ソリューション研究開発室
Solution Research & Development Center, FUJITSU LABORATORIES
†† 富士通研究所コンピュータシステム研究所知能システム研究部
Intelligent Systems Laboratory, Computer Systems Laboratories, FUJITSU LABORATORIES
††† 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University

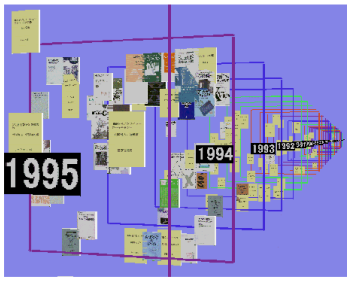


図 1 3 次元ブラウジング空間の例
Fig. 1 Example of 3D broushinng space.

Galaxy of News システム[3]がある．これらのシステムでは，クラスタや概念関係を抽出する必要があり，現在の技術では実用化することが困難な問題を含んでいる．これに対処するため，クラスタや概念関係については利用者の判断に任せ，なるべく簡単に効率的なブラウジングを可能にする方法を提案[4]した．この方法は検索をブラウジングの観点から見直したものである．ここで取り扱うブラウジング法は，情報を代表する複数の特徴（特徴表現と呼ぶ）を選び，それらに対応する値を要素とする特徴ベクトルを作成し，それらの間の類似度を基に情報を 2 次元空間に分布させ，これに一次元のナビゲーション軸（たとえば，時間軸）を加えて作成した 3 次元ブラウジング空間を利用するものである．**図 1** は文献をキーワードの類似度と時間軸を使って作成した 3 次元ブラウジング空間の例である．この方法は 1,000 件程度の情報のブラウジングまでは現在のエントリクラスのパーソナルコンピュータで実用的に利用できる速度のものであった．しかし，10,000 件程度の情報に対しては困難であった．この問題を以降で述べる方法により解決した．2 次元空間に情報を分布させる方法は，特徴ベクトルとその間に定義された類似度を基に，類似する情報は近くに類似しない情報は遠くになるように 2 次元空間上に配置するものである．一般的には，特徴ベクトルの次元は 2 次元より大きいために，2 次元空間に情報を類似度に従ってマップ化する必要がある．この手法としては MDS 法や因子分析などの方法もあるが，直感的に分かりやすく，適用範囲も広い Kohonen の自己組織化マップ（SOM と呼ぶ）[5]を利用した．この手法は WEB のホームページを所属するキーワードの類似性によって 2 次元分布させる例[6]やビデオカットを 2 次元空間に分布させる例[7]など多数の応用が報告されている．しかし，これらの手法は情報の数と特徴表現の数が多いと時間がかかるという問題があった．

そこで，Kohonen の自己組織化マップは汎用であるため速度も遅くまた情報ブラウジングの特性を十分利用していないと考えられるため，下記の 4 つの手法でこの改良を行った．

(1) 特徴ベクトルの要素を決める特徴集合をできる限り情報を分散させるように決定する（前論文[8]参照）．

(2) 学習範囲の削減に応じて情報の最適位置の探索範囲を削減する．

(3) すべての情報を対象としないで最大で 1,000 件程度の適合する情報のみで自己組織化マップを計算し，すべての情報に対応するマップを補間処理により計算する．

(4) 情報が重なって配置されるマップの部分は周囲に分散させることにより配置を均一にする後処理を採用した．

(1) と (4) はブラウジングを見やすくすることに対応し，(2) と (3) は速度の向上に寄与するものである．(2) によって典型的な例で 1/5 程度の計算時間になる．また，計算時間は情報数の 2 乗に比例するため (3) によって 10,000 件の場合には計算時間が 1/100 になると予測されるが，実験でもこのことが実証された．

これらの工夫により，エントリクラスのパーソナルコンピュータでも 10,000 件の情報のブラウジングを可能にできる．このブラウジング手法をインターネット上の情報のように 10,000 件をはるかに超える大量情報に適用する場合やブラウジングの対象である情報の集合に対して特徴ベクトルの要素に対応する適切な特徴表現が選択できない場合には，キーワード検索などの検索キーを指定する検索によりブラウジングの対象である情報の集合を絞り込む必要がある．その場合は，キーワードなどの検索キーを入力し，検索した結果の情報の集合に対してブラウジングをすることになる．

2. SOM の速度改善

2.1 SOM の実装と速度改善

ここでは Kohonen の論文[5]に基づいた SOM の実装法を述べ，速度を改善するため学習範囲，学習回数などのパラメータを決める方法を説明する．

SOM は以下のような情報を代表する特徴ベクトルを入力とする．たとえば，情報として科学技術文献を考え，キーワード「生物」で検索した結果の文献集合に対して適用する場合を考える．この集合を記述するキーワードとして，次のような 9 つのキーワード「エネルギー」，「タンパク」，「遺伝子」，「研究」，「細胞」，「植物」，「進化」，「変化」，「構造」が選択されているとする．各キーワードの出現頻度を要素とする次のようなベクトルを，各文献に対応する特徴ベク

トルと呼ぶ．
A 文献：(11,0,0,2,0,0,2,0,0)，
B 文献：(0,0,3,0,5,1,2,0,0)．
これらの特徴ベクトルは次元が高く，このままでは類似度が分かりにくい．SOM は，特徴ベクトルをそれらの間に定義された類似度を反映する形で，より次元の低い多次元空間上に整列させる方法である．一般的には，この多次元空間として，視覚的に分かりやすい2次元空間が使われる．以下では2次元空間を2次元の正方格子とし，各格子に配置されたマップベクトルに整列結果を学習させるものとして説明する．

学習は，次の2段階で行われる．1）特徴ベクトルに最も類似度の高いマップベクトルを選択する．2）この選択したマップベクトルから2次元空間上で一定の範囲にあるマップベクトルを特徴ベクトルに類似するように変化させる．この変化の割合を学習回数の増加および選択したマップベクトルからの距離の増加に従って減少させ，さらに，変化させるマップベクトルの範囲も学習回数の増加に従って減少させることによって，学習を収束させていく．この学習の過程は，変化させるマップベクトルの範囲の大きさによって大域的な構造を決める初期学習と局所的な構造を決める調整学習に分かれる．この過程をマップベクトルの変化が一定値以下になるまで繰り返すことによって，特徴ベクトルの集合を類似度に従って2次元空間上に配置することができる．

自己組織化マップの実装法はいろいろなものが考えられる．Kohonen[5),9)]には，学習式の形，そのパラメータの条件，マップの初期値の例，学習のための入力ベクトルの選択方法の例，初期学習と調整学習の割合などの条件が述べられているのみで，具体的な組合せ，値は試行錯誤に決めることになっている．しかし，リアルタイムに処理するためには，これらの値を事前に決めておく必要がある．ここでは，類似度を内積として，この章の以降に述べる方法で実装した．その結果，次章で述べるように，学習時間を短縮できることが分かった．

前提条件として，**図2** のような $L \times L$ の正方格子を持つ2次元空間を考える．ここで，L は，n を入力情報数とすると，各格子に1つずつ情報を配置するために $\sqrt{n}$ を超える最小の整数とする．各格子点 (i,j) に存在するマップベクトル (m_{ijk}) を学習させることを考える．k は特徴表現を表す．また，各入力情報の特徴ベクトル (x_k) は $\sqrt{\sum_k x_k^2} = 1$ に正規化されているものとする．

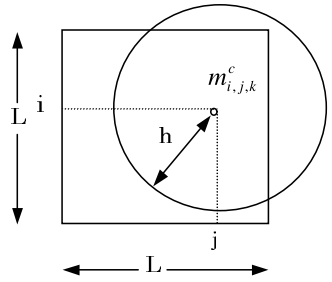


図2 マップと学習範囲
Fig. 2 Map and range of learning.

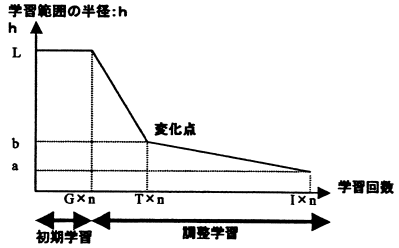


図3 学習回数に対する学習半径の変化
Fig. 3 Radius of range of learning versus number of learning.

(1) 類似度 $s(x,y)$ は内積 $\sum_k x_k \times y_k$ で表現し，この類似度により入力情報の特徴ベクトル (x_k) に最も類似度の高い格子点 l_c を選択する．

(2) Kohonen[5)] の示唆に従って学習回数と学習範囲の関係を**図3** のグラフの形で近似し，これで決まる半径 h の範囲内の格子点のマップベクトルを以下の式により入力情報の特徴ベクトルに類似するように更新していく．t は学習回数で，r は学習対象の格子点と l_c との間のユークリッド距離で，α は正の係数である．

$$m_{ijk}(t+1) = m_{ijk}(t) + c(t)e^{-\alpha r^2} (x_k - m_{ijk}(t))$$

ここで，学習は入力情報全体を繰り返し適用することにより進める．図3 の G は初期学習の終了点の n を単位とした学習回数（n 単位学習回数と呼ぶ）であり，T は調整学習の緩やかに変化させる範囲の開始点の n 単位学習回数である．β は1より小さい係数である．

$c(t) = e^{-\beta}$
$(t < G \times n$：図3 の初期学習の範囲の場合)
$c(t) = e^{-\beta} \times G \times n/t$
$(t \geq G \times n$：図3 の調整学習の範囲の場合)

後の実験では $\alpha = 1/(\text{学習範囲})^2$ とし，$\beta = 0$ とした．

(3) $m_{ijk}(t+1)$ を正規化し，それをその格子点の新しい特徴ベクトルとする．

(4) (1)から(3)を入力情報全体を単位として，各入力情報の特徴ベクトルに対して実行する．その中で，各特徴ベクトルに最も類似度の高い格子点のマップベクトルとその特徴ベクトルの類似度の最小値を求める．この値の n 単位学習回数に対する変化が一定の値以下になるまで繰り返す．たとえば，収束条件として前回との最小値の差が有効数字の上3桁が同一になるまで実施する．

(5) 収束した2次元空間の格子点のマップベクトルを使い，各入力情報の特徴ベクトルに最も類似度の高いマップベクトルを持つ格子点にその入力情報を配置する．また，座標として採用した特徴表現を単位ベクトルとして考え，同様の位置の計算を行い，2次元空間に配置する．

以上の実装法の中で，速度向上の可能性があるところは以下の点である．

(1) 入力情報の特徴ベクトルのマップ上の最適位置を探索する処理：通常は n に比例する回数の内積計算が必要であるが，これを削減することが考えられる．これについては n 単位学習回数に対して各入力情報の学習位置の変化を調べることにより，削減する方法を検討することにした．計算時間とパラメータの関係は以下のように近似できる．

$$\text{計算時間} \propto L^2 \times I \times n \tag{1}$$

(2) マップベクトルの学習処理：図3のグラフをx軸を中心に回転させた回転体の体積に比例するベクトルの修正計算を削減することが考えられる．これは，図3のグラフとx軸の間の面積を小さくするようにパラメータ（G, T, a, b, I）を決めることにより実現することにした．これらのパラメータと計算時間の関係は以下の式で近似できる．

$$\begin{aligned}\text{計算時間} \propto\ & (L^2 \times G \times n + 1/3 \times (T-G) \\ & /(L-b) \times n \times (L \times n - b^3) \\ & +1/3 \times (I-T)/(b-a) \times n \times (b^3 - a^3))\end{aligned} \tag{2}$$

式(1)と(2)および，L^2 は n に比例することから，I と G，その次に $T-G$ が n^2 に比例する項の係数であるため計算時間に大きく影響することが分かる．マップベクトルの学習処理が入力特徴ベクトルのマップ上の最適位置の探索に影響するため，2.2節で学習処理の短縮を説明し，2.3節で最適学習位置の算出時間の削減について説明する．

ここで2次元空間の分布を評価するために以下の2つの評価指標を説明しておく．

(1) 忠実度：これは，SOM_PAKでは入力ベクトル x_{ik} とマップの各格子の最適位置のマップベクトル m_{ik}^c との距離の平均値を量子化誤差と呼んでおり，結果の評価に利用している．これは入力ベクトルに対する忠実度を表すので，ここでは忠実度と呼ぶ．以下のような式で記述できる．

$$\text{忠実度} = \sum_{i=1}^{n} \sqrt{\sum_k (x_{ik} - m_{ik}^c)^2}/n$$

(2) 均一度：ブラウジングの観点から分布の均一性を評価する方法として以下のような均一度[8]を考えた．これは「マップの各格子点に1つ配置される状態を理想として，それからのずれを表す」ものと考え，以下のような式で表現した．そこで，$n(i,j)$ は位置 (i,j) に配置された入力情報数，n は全入力情報数である．

$$\text{均一度} = \sum_{i,j} (n(i,j) - 1)^2/n$$

この値は，全格子数が入力情報数に等しいとすると，各格子に1つずつ配置された場合には0になり，格子の半分に2つずつ配置された場合には1になる．この値は0に近い方が分布は均一だと考えられる．

2.2 学習処理の短縮

図3の形に計算範囲を制御し，これに合ったマップベクトルの初期値と各パラメータの選択による効果を測定した．テストデータは科学技術図書1,095件を使用し，必要に応じて岩波新書データ1,211件を利用した．計算では，断らない限り，100件以上ヒットするキーワードそれぞれ，35個，12個による検索結果である情報の集合に対する結果を平均した値を算出した．

マップベクトルの初期値としては，入力データの特徴ベクトルをランダムに配置するランダム配置，入力順にマップの左上から配置する入力順配置，中心に1つだけ配置する中心配置の3種類の方法で実験したが，あまり差がないため，入力情報の順序を1回だけランダマイズし，最初の1つをマップの真中に配置することにした．

図3のパラメータG，T，I，a，bを，次の3つの考え方に従って決めることにした．1）Kohonenの指摘に従う．2）計算量が少なくなるように論理的な最小値にする．3）SOMの計算量への影響度合いを考慮する．

これらの考え方で決めた値を確認するため，第1に収束性で確認した．これは，まず学習が収束することが前提であり，その次に，Kohonenが学習の良否に利用している忠実度で判断した．均一度については，

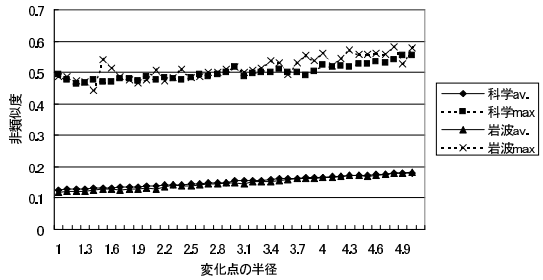


図 4 変化点における学習半径と非類似度
Fig. 4 Radius of range of learning and similarity distance at inflection point.

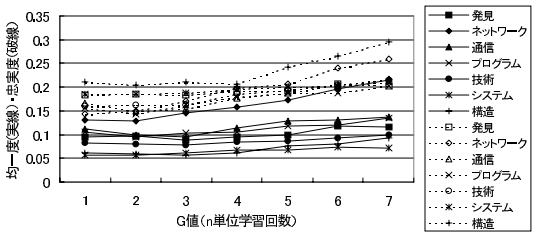


図 5 G の変化に対する均一度と忠実度
Fig. 5 Radius of range of learning and similarity distance at inflection point.

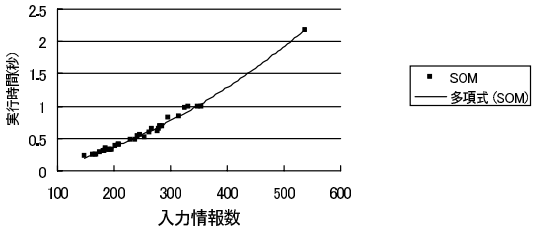


図 6 入力情報数に対する実行時間の変化（Pentium II 400 MHz）
Fig. 6 Execution time versus number of data (Pentium II 400 MHz).

応用の観点からの評価値であるので，汎用性を考え，なるべく収束性と忠実度で決めることにした．しかし，計算量が大きく変わる部分（G）のみは，均一度の観点からの評価も実施した．

収束性は，非類似度を 1 − 類似度 と定義し，入力情報全体の中での，最適学習位置のマップベクトルと入力情報の特徴ベクトルの間の非類似度の最大値の変化により測ることにした．1）の観点からは，a = 1 を決めた．これは最近傍の 4 つの格子点のみが範囲に入ることを意味する．次に，2）の観点から，学習回数を減らすことを考え，$G = 1$，$T = 2$ を基準として試行錯誤した．

T の値については，b との関係で調べることにした．$b = 1.0$，2.0，3.0 の 3 点で T の値による収束性を調べたが，$T = 2$ から 7 の範囲の間では差があまり出なかったので，T は計算時間の点から最小の $T = 2$ を採用することにした．b の値を決定するために，学習範囲の大きさと，マップベクトルと，そこに配置された入力情報の特徴ベクトルとの非類似度の平均値と最大値の関係を調べた．科学技術図書 1,095 件と岩波新書 1,211 件に対して 100 件以上の検索結果となる集合 35 集合と 12 集合に対する平均値を算出した．**図 4** は b の値を 1.0 から 5.0 まで 0.1 単位に非類似度の平均と最大を算出しグラフにしたものである．非類似度の平均値は増加するため，最大値の傾向により，b の低い値の部分で決めることにした．最大値は，b が 1 から 1.4 まではほぼ同じような値になっている．これはこの範囲に入る格子点が 4 つの場合に対応する．

G の値については，$G = 1$ から 7 の範囲では収束性については差が出なかったが，n 単位学習回数の最後の学習での非類似度の最大値を調べると，$G = 1$ から 7 の間での最小値が $G = 1$ から 4 の間にあった．この結果から，学習回数の最も少ない $G = 1$ で計算しても大きな値の差はないことが分かった．

最後に，I の値は大きければ大きいほど忠実度が良くなるが，$G = 1$，$T = 2$ の場合には 9 回程度で減少率が変化し緩やかになる傾向が分かり，配置結果も大きな変化もなかった．また，G や T の値の増加に対応させて I も増加させる方がよいことも分かった．しかし，配置そのものは，I の変化には大きく影響されることはなかった．そこで，以降では $G = 1$，$T = 2$ の場合には $I = 9$ にした．

以上のパラメータは主として収束性に忠実度を加味して決定したが，均一度についても確認のためテストを行った．G の値が計算速度と均一度に最も影響することが考えられるので，G に対する変化をテストした．科学技術図書データを 7 つのキーワードで検索した結果の 7 つの図書の集合に対して計算した結果を **図 5** に示す．これから均一度も忠実度も $G = 1$ から 4 の間に最小値があることが分かる．

以上の結果，$G = 1$，$T = 2$，$a = 1$，$b = 1.4$，$I = 9$ を採用し，自己組織化マップの計算時間を測定した．測定環境として，Pentium II 400 MHz，128 MB メモリ，WindowsNT を利用した場合に，科学技術図書データに対して，入力ベクトルの要素数 9 の場合の計算時間の結果を**図 6** に示す．この結果では 537 件で 2.17 秒であった．自己組織化マップは $O(n^2)$ であるが，係数を削減することにより 1,000 件程度（約 4 秒）まではリアルタイムに計算可能な範囲になったと考えられる．

SOM の速度を評価するため，Kohonen グループが

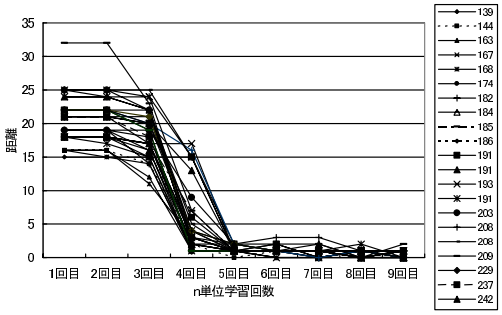


図 7 学習回数に対する最適位置の最大移動距離の変化
Fig. 7 Maximum drift of the best position versus number of learning.

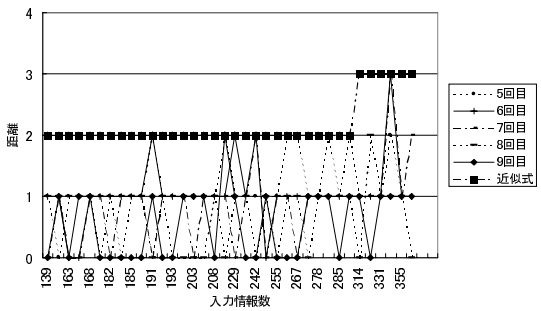


図 8 入力集合の情報数に対する最適位置の最大移動距離の変化（5 回目以降）
Fig. 8 Maximum drift of the best position versus number of data.

配布している SOM_PAK Version3.1[9]と比較してみた．CPU が Pentium II 400 MHz，WindowsNT4.0 で 735 件の入力情報，マップの大きさ 28×28，学習回数 6,615 で，他のパラメータも同等にして速度を測定した．SOM_PAK は 9 秒かかり，提案の方法は 1.973 秒で SOM_PAK の約 22%の時間で計算できることが分かった．

2.3 最適学習位置の算出時間の削減

これをさらに高速化するため，入力情報の最適な学習位置を決める場合に全マップの格子上のマップベクトルを調べるのでなく，範囲を絞ることができないか調べてみた．そこで，n 単位に前回の最適位置と新しく計算した最適位置の距離を算出し，格子間の単位で距離の分布を見てみた．**図 7** は入力集合の情報数に対して n 単位学習回数に対する最適位置の移動の最大距離をグラフ化したものである．3 回目までは大きな変化があることが分かった．この最大値は 4 回目から 5 回目にかけて急激に小さくなり，5 回目以降は一定値におさえられることが分かった．図 7 は図 3 にほぼ対応していることが分かる．**図 8** は 5 回目以降の詳細な変化を見るものである．これを見ると最大値は入力情報数に少し依存しており，学習回数が増加するとともに少し大きくなる傾向がある．この図から，5 回目以降の最適位置の計算は，距離を (入力集合の情報数)/300 + 2 で近似することにした．1 回目から 3 回目まではマップの対角線の長さとし，4 回目は，直線で 3 回目と 5 回目の距離の間を結んだものにした．この結果，式 (1) は以下のようになる．

$$\text{計算時間} \propto (L^2 \times (T+1) \times n + (L^3 - (n/300+2)^3)/(3 \times (L - n/300 - 2) + (n/300+2)^2 \times (I - T - 2) \times n) \quad (3)$$

この効果を上述の測定と同じもので測定した結果，1.73 秒となった．この処理を導入しない場合の 88%，SOM_PAK の約 19%となった．さらに，確認のために，1,000 件のデータで学習回数との関係から SOM_PAK と比較したのが**表 1** である．この表から分かるように提案方式は学習回数の増加に対する時間の増加が SOM_PAK に比べて低くなっている．学習回数の多い方が分布の安定性が増すので，現在は入力情報数の 9 倍の学習回数を利用することにしている．また，表 1 で示すように，忠実度も提案方式が SOM_PAK の約 1/2 になっていることが分かる．理由は SOM_PAK 内のアルゴリズムを確認する必要があるが，図 3 と同様な形の学習をすると仮定し，全体の学習回数を一定とすると，G や T に対応する値の増加により，忠実度が悪くなることが考えられる．

表 1 提案方式と SOM_PAK の速度および忠実度
Table 1 Execution time and fidelity of the proposed method and SOM_PAK.

学習回数	実行時間 提案方式	実行時間 PAK	忠実度 提案方式	忠実度 PAK
4,000	2.824 秒	7.03 秒	0.0278	0.0538
5,000	2.845 秒	8.56 秒	0.0241	0.0517
6,000	2.864 秒	10.49 秒	0.0222	0.0503
7,000	2.875 秒	12.19 秒	0.0211	0.0495
8,000	2.895 秒	14.17 秒	0.0203	0.0487
9,000	2.915 秒	15.93 秒	0.0197	0.0483
10,000	2.925 秒	17.63 秒	0.0192	0.0477

注）PAK は SOM_PAK を示す．

3. 大量情報への対応

10,000 件以上の大量情報に対する対応策として，ブラウジングを考慮したランダムサンプリングによる作成法を述べ，実験により，その効果を確認する．

自己組織化マップの計算時間は入力する情報数の 2 乗になる要素があるため，入力情報数の多い場合には計算時間が非常に長くなる．上述の方法では 1,000 件で Pentium II 400 MHz で 5 秒程度であるため，今後の計算機の高速化を考えても，計算できる入力情報数

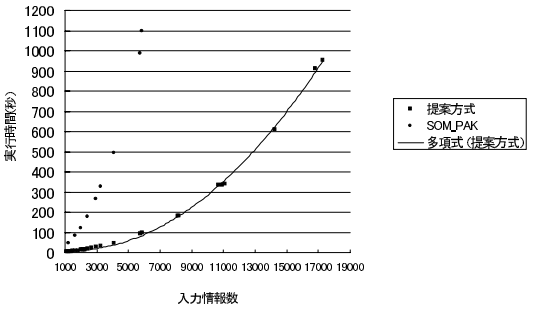


図 9 ホームページデータの場合の実行時間
Fig. 9 Execution time in the case of home page data.

は数千件までと考えられる．たとえば，**図 9** のホームページデータに対する Pentium II 400 MHz での測定結果から分かるように，SOM_PAK に比較するとよいのだが，それでも 1,800 件程度（約 10 秒）以上はリアルタイムには難しくなる．そこで，10,000 件に対応するために，ランダムサンプリングにより 1,000 件程度抽出し，マップを計算することにした．ランダムサンプリングの結果を利用する方法として 2 通りの方法が考えられる．以下にそれらの方法と SOM の計算時間の主要部分（式 (1)，(3) の n^2 に比例する部分）について述べる．ここで，s をサンプリング数とする．

1）マップの大きさはそのままに，学習の対象であるデータ数にのみ反映する方法．

計算時間 $\propto c_1 \times n \times s$

2）マップの大きさも，学習対象の入力情報数も s にする方法．この場合は，マップの拡大処理が必要で，以下の計算時間評価式の第 2 項が増える．

計算時間 $\propto c_1 \times s \times s + c_2 \times n$

現在，考えている n，s の値は 10,000 と 1,000 であり，係数の c_1 は内積，c_2 は内挿式の計算の程度である．これらの係数が同じオーダの計算時間だとすると，1）の方が，2）より 9.9 倍時間がかかることになる．この結果，2）の方式を採用することにした．

これまで，クラスタリングを利用したブラウジング手法[2),10)]では，大量情報に対応するために，ランダムサンプリングが利用されてきた．これらの手法の特徴は，クラスタ数との関係で，サンプリング数を決定することにあった．ここで提案するブラウジング手法ではクラスタを見つけることを目的にするのではなく，入力情報間の類似性により空間配置を決め，一覧性の向上によりブラウジングを効率化しようとするものである．これは，多種多様な入力情報では事前にクラスタ数を決めるのことは困難であり，分類より，情報間の関係を概観する方が良いと判断したことによる．そのため，クラスタ数は決める必要がなく，通常のランダムサンプリング手法に，入力情報の配置に役立つかどうかという観点から入力情報の特徴ベクトルを分類し，それに従って，抽出の優先順位を付けた手法を考えた．

3.1 標 本 抽 出

ここで対象としている情報には，テキスト，静止画像，音情報など様々な情報があるため，それらの特徴とそれに対する特徴ベクトルはブラウジングのための分布空間作成に適切でないものも種々混じる可能性がある．これらには次のようなものが考えられる．

(1) 零ベクトルは SOM でマップ上の最適位置を決める場合に類似度の内積が零になり最適位置を決めることができない．そのため分布の決定に役立たなくなる．

(2) 単位ベクトルは SOM で，マップ上の最適位置を決めることができるが，学習のときに，1 のベクトル要素以外の要素に対して平等に値を小さくするのみで，これらの要素に対して差をつけることがない．要素は特徴に対応しておりブラウジング時の指標になるが，単位ベクトルは特徴間の関係を配置に反映するのに役立たない．

たとえば以下のような例がある．

図書の集合を考える．「情報検索」，「書誌情報」，「シソーラス」，「一次情報」，「二次情報」，「論理式」，「画像情報」の 7 つのキーワードが特徴ベクトルの要素として選択された場合，各キーワードが図書データに出現する頻度でベクトルが表現されているとする．特徴ベクトルの要素とその数は，情報の集合の要素である図書から決定することになる．特徴ベクトルの要素数は計算時間および一読性の観点から 1 桁または 2 桁の数にする必要があるので，上記 7 つのキーワードの選択によっては，情報の集合の要素である図書にそれらのキーワードが 1 回も出現しない図書が生じる可能性がある．たとえば，タイトル「原価計算システム」という本が入力情報の集合の中にあるとすると，その中には上記 7 つのキーワードがどれも 1 回も出現しない可能性があり，零ベクトル (0,0,0,0,0,0,0) になることが考えられる．この場合には，類似度としてコサインを利用した場合には，2 次元空間に配置できないことになる．これは，このベクトルでは，どの図書に類似するか判定できないことを意味する．また，この本に「情報検索」というキーワードのみがある場合には正規化すると (1,0,0,0,0,0,0) という単位ベクトルになり，他の要素との関係がないため，SOM の計算では，2 つ以上の要素（特徴）の関連性を決めるのに役立たない．

表 2 零ベクトルと単位ベクトルの数
Table 2 Number of zero and unit vectors.

特徴ベクトル	零ベクトル	単位ベクトル	その他のベクトル
16 個の分類 KW	0	6,385	1,309
25 個の自動抽出 KW	300	2,936	4,798

注）KW：キーワード，自動抽出：修正均等分割法

表 3 標本抽出拡張法の効果
Table 3 Effect of method of sampling and expanding.

入力情報	処理方法	作成時間	均一度	忠実度
画像	全情報利用	171.13	0.0430	0.0489
画像	標本抽出拡張	4.67	0.0408	0.0626
特許文	全情報利用	1242.91	0.0401	0.3153
特許文	標本抽出拡張	12.55	0.0585	0.3775

注）要素数は画像が 9，特許文が 16 である．

具体的な例として，小学館から実験用にお借りした静止画像データの説明文（7,674 件）を対象に調べてみたところ**表 2** のような結果となった．特徴ベクトルの要素であるキーワードの選択法は，すべての説明文に入っている分類キーワードを選択する方法と，修正均等分割法[8]である．表 2 の場合のように，選択したキーワードが入力情報の中に 1 つしかない場合が多いような状態になることがよくある．特に，表 2 の分類キーワードの場合のように単位ベクトル数が多く（1,000 件を大きく超える場合），SOM を利用するときにランダムに最大数分（1,000 件）を抽出すると，キーワード間の関連が配置にうまく反映されない場合がある．この場合には，キーワード間の意味的関連性を特徴ベクトルに導入する方法も考えられるが，これは事前に意味的な関連情報を作成する必要があり，入力情報が可変で大量にある場合にはコストがかかるという問題が生じる．これを解決するために，SOM の配置は，特徴ベクトルの非零の要素数が 2 以上のものから優先してランダム抽出することにした．これにより，妥当な配置になる可能性が増える．

3.2 マップの拡大

抽出した情報で計算したマップを全情報に適用するためにはマップを拡大する必要がある．SOM の結果のマップは，多次元ベクトル空間を 2 次元ベクトル空間にマッピングしたものである．そのため，結果のマップにはある程度の誤差を含んでいる．そのため，厳密な計算は必要がないと判断し，計算時間の少ない手法で実現することにした．以下では，抽出数を 1,089 件とし，33×33 のマップを基準として拡大法を説明する．

1）入力情報の数が 1,089 までは，そのまま SOM を利用し，これ以上の数は，1,089 個の入力情報を上述の方法で抽出し，まず 33×33 の格子上のマップを SOM で算出する．

2）入力情報数 n から算出した目標の格子の一辺を L とし，x 方向に 33 の格子を L の格子に両端を揃えて変換する．具体的には，L 個の格子点上のマップベクトルを 33 の格子点上のマップベクトルから直線補完により算出する．これを 33 回（y 方向）繰り返すことになる．

たとえば，基準のマップの i 番目と $i+1$ 番目の格子点の間に拡大マップの i' 格子点があるとすると，i' 格子点の特徴ベクトルは以下のように算出できる．

$$m'_{i',j,k} = (m_{i,j,k} r_{i,i'} + m_{i+1,j,k} r_{i',i+1}) / (r_{i,i'} + r_{i',i+1})$$

ここで，$r_{i,i'}$，$r_{i',i+1}$ はそれぞれ i と i' 格子点および i' と $i+1$ 格子点間の距離を表す．

3）$L \times 33$ 個の格子点上のマップベクトルから，y 方向に 33 の格子を L の格子に両端を揃えて 2）と同様に変換する．これを L 回（x 方向）繰り返すことになる．

以上の計算により，1,089 より大きい数の入力情報数の場合のマップベクトルを計算することができる．

3.3 実験評価

マップの作成時間を比較するために，12,169 件の特許広報文と 7,674 件の画像データ（小学館より借用）に適用してみた．特徴ベクトルは，画像データが色相ベクトルで要素数が 9，テキストデータはキーワードベクトルで要素数が 25 である．コンピュータは Pentium II 400 MHz，128 MB メモリ，Windows2000 を利用した．測定時間は，ランダムサンプリングから SOM による配置までの時間を測定した．これ以降，ランダムサンプリングし，マップを拡大する方法を標本抽出拡張法と呼ぶ．

結果は**表 3** にまとめた．実行時間は元の時間の画像が 2.7%，テキストが 1.0%までに減少した．均一度は，画像はほぼ変わらないが，テキストの場合には 45.9%悪化している．忠実度は画像が 53.5%，テキストが 19.7%悪化しているが，配置された結果の空間を目視したレベルでは問題になる配置にはなっていない．おそらく，大域的な配置は問題のないレベルであり，ブラウジングの目的には十分利用できるものと考える．また，値そのものの最大値は格子間隔の半分以下であるので，どちらのデータでもブラウジングの観点からは問題のないレベルであると考えられる．キーワードベクトルの均一度の悪化が大きい理由としては，ランダム抽出率の差が大きい点が効いており，逆に忠実度

の悪化が少ないのは選択したキーワードの頻度の差が大きくないため，キーワードベクトルのパターンの差が少ない点が効いているものと考えられる．均一度の改善については次に述べる．

4. 多重配置の解消

入力情報の多重配置はブラウジングのときの一覧性を下げるために，解消する必要がある．入力情報の多重配置が見つかった場合に，一定の範囲に空がある場合には多重配置している情報をその場所に配置することにより多重配置を解消することが考えられる．また，配置をゆがめて隣り合った入力情報間には矛盾があまりないように配置することも考えることができる．ここでは，このような観点から多重配置を解消する方法について述べる．

多重配置解消を以下のように考えた．

条件1）ブラウジングの観点から，最初に配置された位置（最適配置位置と呼ぶ）から，ある一定の範囲内（半径 R の円内）にのみ配置する．これにより，探索範囲が限定される．

条件2）情報が配置されていない格子点で，その特徴ベクトルとマップベクトルとの間の類似度がある一定以上の場合に配置する．これは，マップベクトル $m_{i,j}$ と情報のベクトル v との類似度 $s(m_{i,j}, v)$ と，マップの一辺の格子数 L とベクトルの要素数 n_v の関数 f を使って，$s(m_{i,j}, v) > f(L, n_v)$ と表現できる．この条件では，忠実度の大きな悪化を防ぐ効果がある．

条件3）他の情報がすでに配置されている格子点で，配置しようとしている情報の特徴ベクトルとマップベクトルとの間の類似度が，既存の情報の特徴ベクトルとマップベクトルとの類似度より高い場合に配置する．置き換える位置に2つ以上の情報がある場合には類似度の最も低いものを次の再配置処理の候補とする．この条件は忠実度の大幅な悪化を防ぎながら，分布を均一化する効果がある．この条件は格子位置 (i, j) にある多重配置されている情報の集合 $W_{i,j}$ とその情報の特徴ベクトル w_k により $s(m_{i,j}, v) > \max_{w_k \in W_{i,j}} s(m_{i,j}, w_k)$ のように表現できる．

具体的には，多重配置している情報を1つ選択し，多重配置位置を中心に近い配置位置から**図10**に示す順にこれらの条件を適用していく．条件を適用する方法は，以下のようになる．

1）再配置の対象位置が空いているがどうかを調べる．

2）空いている場合には，その位置のマップベクトルと配置しようとしている情報の特徴ベクトルの類似度が一定以上の場合に配置し，次の多重配置情報の処理に移る．低い場合には次の位置へ移動する．

3）空いていない場合には，マップベクトルとの類似度の比較を実施し，類似度が低い場合には次の位置へ移動し，高い場合には，その位置にすでに配置されている情報のうち，最も特徴ベクトル間の類似度の低い情報を選び，それの再配置を実行する．

4）置き換えにより再配置を行う場合には，入力情報の初期配置位置からの図10における現在の配置位置の順番を求め，その順番の次から1）の処理を適用していく．

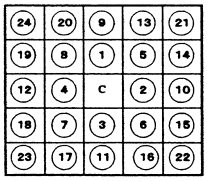


図10 再配置適用順
Fig. 10 Execution order of reallocation.

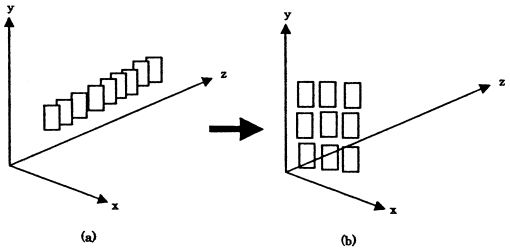
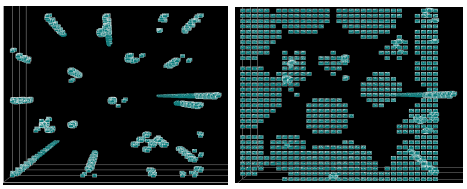


図11 多重配置解消法の原理
Fig. 11 Principle of dissolving overlap.

これを簡単な例で説明する．**図11**の(a)に示すように，ある1つの格子に多数の情報が配置された場合，現在は，その位置を中心に，情報を発見しやすいように螺旋状に配置している．この例では一列に並べている．この場合に，図11の(b)にあるように，周りの格子に情報が配置されていない場合には，近い位置から順に周りに配置していくことができる．もし，周りに他の情報が配置されていた場合には，どちらがよりマップの格子ベクトルに類似するかによって，入れ替えを行う．原則として，情報の再配置位置は最初に最適な位置（マップベクトルとの類似度が大きい位置に配置）にあるので，その位置からの距離で制限する．また，マップベクトルとの類似度もあまり低いと周りとの関係で探索の効率が悪くなるので，制限する必要がある．**図12**は1,050枚の写真データのタイトルと写真家名のデータをテキストデータとしてキーワードで情報を配置した場合の再配置前(a)と再配置後(b)を示したものである．

以上の多重配置解消方法の各条件がどの程度効果があるかを調べるため，実験を行った．条件 1）と 2）を実行した場合をケース 1 とし，条件 1）と 2），3）を実行した場合をケース 2 とし，計算負荷が高いと思われる条件 3）の効果を明確にすることにした．測定は，SOM 実行後とケース 1 とケース 2 の実行後に分けて測定を行った．パラメータは以下の考え方で決めた．
条件 1）の再配置の範囲は半径 R を次の式で決めた．$R = L/\sqrt{n_v} \times 3/4$ これは特徴がマップ上に等間隔に配置されるとした場合の半径の 1.5 倍にあたる．
条件 2）の類似度の境界値 s_t は多重数がデータ数の 1/3 を超える場合と超えない場合に分けて，以下のように決めた．
1/3 を超える場合：$s_t = \gamma - (n - n_0/3)/(2n/3)(\gamma - 0.5)$ ここで，γ は類似度の境界値の基準値で 0.5 以上の数値とする．これは多重配置数が多い場合に条件を緩めて一覧性を重視するためのものである．以下の実験では 0.7 を使用した．n_0 は n から零ベクトル数を引いた値を示す．



(a) (b)

図 12 再配置例
Fig. 12 Example of reallocation.

表 4 実験で利用したパラメータ一覧
Table 4 Parameters in the experiment.

データ名	件数	特徴表現	要素数	最大半径
写真画像	1,050	色相ベクトル	9	14
画像	7,674	色相ベクトル	9	16
特許広報文	12,169	キーワード	25	16

1/3 以下の場合：$s_t = \gamma$

実験ではマップの拡大を利用しない場合の効果を確認するための 1,050 枚の写真データと，前節で利用した 7,674 件の画像データと 12,169 件の特許広報文を使用した．利用したパラメータ値を**表 4** に，結果を**表 5** にまとめる．

この結果より，多重配置数は 1/2 から 1/4 に減少し，均一度は 29%から 55%の改善がみられるが，忠実度は悪化する．特に，画像の場合には元の値が良いために 2〜4 倍程度の悪化が見られるが，値としてはテキストの場合より良い値となっている．ケース 2 は全体的に時間はかかるが，分布を押し広げる処理のため忠実度の悪化が少ない点が優れている．一方，処理時間と多重配置数を重視するなら，ケース 1 の処理が良いことがいえる．また，入力情報数が多く多重配置数が多い場合にはケース 2 の方が均一度の改善率が良いようである．これは，空き位置の利用が分布を押し広げる効果により多く起こる可能性が高いからと考えられる．反対に写真画像の場合には多重配置数が少ないため空き位置も少なく，空き位置に到達する前に置き換え処理が入り，最終的に空き位置を利用できない可能性が増えるからだと考えられる．

これらの結果より，この多重配置解消処理は，情報の分布を均一に押し広げることによりブラウジングの効率を向上させる方法であることが分かる．

今回，パラメータはメディアが変わっても同じものを利用したが，画像の場合とテキストの場合の特徴ベクトルの性格が異なるため，さらに，最適なパラメータをメディアごとに決めることにより改善することが可能と考える．

5. おわりに

10,000 件を超す大量情報も現時点のエントリクラスのパーソナルコンピュータでブラウジング可能なこ

表 5 多重配置解消の効果
Table 5 Results of dissolving overlap.

データ名	処理過程	多重数	多重割合	均一度	忠実度	処理時間
写真画像	SOM	452	0.4305	0.0359	0.0301	3.54 秒
	ケース 1	69	0.0657	0.0171	0.1193	3.55 秒
	ケース 2	105	0.1000	0.0194	0.0873	3.56 秒
画像	SOM	6,036	0.7866	0.0408	0.0626	4.7 秒
	ケース 1	1,737	0.2263	0.0224	0.1512	4.8 秒
	ケース 2	1,566	0.2041	0.0180	0.1268	5.3 秒
特許広報	SOM	10,519	0.8644	0.0585	0.3775	12.6 秒
	ケース 1	3,496	0.2873	0.0417	0.4980	13.1 秒
	ケース 2	3,981	0.3271	0.0359	0.4394	14.5 秒

処理時間は SOM 開始時からの当該処理過程終了までの時間

とを示した．10,000件の情報を一覧しブラウジングできることで，利用者にある種の爽快感または新鮮さを与えることができるのではないかと考えている．課題としては，情報全体の概要と絞り込んだ後の部分空間の分布を組み合わせて，連続的に最適な分布空間の下で，ブラウジングが可能なようにすることがある．現在，これらの技術を応用して，ディジタルカメラ写真データの3次元ブラウジングシステムや百科事典用の3次元インタフェースシステムを試作している．

謝辞 実験データを快く利用させていただいた小学館インターメディア部電子百科および小学館ネットコンテンツ部の皆様に感謝いたします．

参 考 文 献

1) Cutting, D.R., Karger, D.R., Pederson, J.O. and Turkey, J.W.: Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections, *ACM SIG-IR '92*, pp.318–329 (1992).
2) Cutting, D.R., Karger, D.R. and Pederson, J.O.: Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections, *ACM SIG-IR '93*, pp.126–134 (1993).
3) Rennison, E.: Galaxy of News, An Approach to Visualizing and Understanding Expansive News Landscape, *UIST '94*, pp.3–12 (1994).
4) Kakimoto, T. and Kambayashi, Y.: Browsing Functions in Three-Dimensional Space for Digital Libraries, *International Journal on Digital Library*, Vol.2, pp.68–78 (1999).
5) Kohonen, T.: The Self-Organizing Map, *Proc. IEEE*, Vol.78, No.9 (1990).
6) Honkela, T., Kaski, D., Lagus, K. and Kohonen, T.: WEBSOM-Self-Organizing Maps of Document Collections, *Proc. Workshop on Self-Organizing Maps* (*WSOM '97*) (Jun. 1997).
7) Hatano, K., Kamei, T. and Tanaka, K.: Clustering and Authoring of Video Shots Using Hybrid-type Self-Organizing Maps, *Proc. International Symposium on Digital Media Information Base*, Nara, Japan, pp.150–158 (1997).
8) 柿元俊博，上林彌彦：ブラウジング検索のための最適な特徴表現選択法，電子情報通信学会論文誌 D-I，Vol.J82-D-I, No.1, pp.130–139 (1999).
9) Kohonen, T., Hynninen, J., Kangas, J. and Laaksonen, J.: SOM_PAK: The Self-Organizing Map Program Package, Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Espco (1996).
10) Guha, S., Rastogi, R. and Shim, K.: CURE: An Efficient Clustering Algorithm for Large Databases, *Proc. ACM SIGMOD Conf.*, pp.73–84 (1998).

(平成13年1月29日受付)
(平成14年1月16日採録)

柿元 俊博（正会員）

1973年京都大学大学院理学研究科修士課程修了．同年，富士通株式会社入社．以来，情報検索，機械翻訳システムの開発，マルチメディア情報検索システムの研究開発に従事．現在，富士通研究所ソリューション研究開発室に所属．

上原 祐介（正会員）

1987年名古屋工業大学工学部情報工学科卒業．1992年名古屋大学大学院工学研究科情報工学専攻博士後期課程単位取得退学．同年株式会社富士通研究所入社．マルチメディア情報検索の研究に従事．人工知能学会会員．

上林 彌彦（正会員）

1965年京都大学工学部電子工学科卒業．1970年同大学大学院博士課程修了．1984年九州大学工学部教授．1990年京都大学工学部教授．1998年改組により京都大学大学院情報学研究科教授，現在に至る．この間，カナダ・マッギル大学，クウェート大学，および中国・武漢大学，スイス連邦工科大学EPFL客員教授．データベース，協調処理等に興味を持つ．情報処理学会理事やACM日本支部副支部長等を務めた．電子情報通信学会米沢賞，日本科学技術情報センター丹羽賞，情報電子通信学会著述賞，ACM SIGMOD貢献賞．情報処理学会フェロー，電子情報通信学会フェロー，IEEEフェロー，ソフトウェア科学会，ACM各会員．