

データ宝石箱 II ~ 位置情報テンプレートを用いた 大規模階層型データのグラフィックスショーケース

山口 裕美 伊藤 貴之

日本アイ・ビー・エム(株) 東京基礎研究所 E-mail: yyumi@trl.ibm.com

大規模階層型データを一望する新しい視覚化手法として、筆者らは「データ宝石箱」を提案している。この手法は、ノードを表現するアイコンを長方形で囲み、その長方形群をさらに大きな長方形で囲み ... という入れ子構造によって階層型データを表現することで、大規模な階層型データの全体を一画面に配置することを実現している。

本報告では、データの画面上の位置を指定したテンプレートを入力し、できるだけそのテンプレートの指示にしたがうようにデータを配置する拡張手法を提案する。提案手法により、従来手法の「ノードの画面上のおおまかな位置関係をあらかじめ指定できないため、類似するデータにおいても非常に異なる配置結果を得ることがある」という問題を解決することができる。本手法は例えば、ユーザーのデザイン意図を反映した階層型データの視覚化、画面上の座標軸に意味をもたせた階層型データの視覚化、時系列に沿って微量ずつ変化する階層型データのシームレスな視覚化、などに向いている。

Data Jewelry Box II: A Graphics Showcase for Large-scale Hierarchical Data Visualization using Templates of Position Information

Yumi YAMAGUCHI Takayuki ITOH
IBM Research, Tokyo Research Laboratory

We have proposed "data jewelry box" algorithm, which provides the overview of large-scale hierarchical data. The algorithm places whole the large-scale hierarchical data in one display space by using nested rectangles. It first packs nodes and encloses by a rectangle, and then repeats the process of enclosing lower-level data by larger rectangles.

This paper presents an extended algorithm that refers templates that describe the desirable positions of nodes of data while packing the data. The presented algorithm solves the problem of existing "data jewelry box" algorithm; it often causes very different layout results from similar data, because it is not able to specify the rough location of nodes. The presented algorithm is especially suitable for design-oriented hierarchical data layout, axis-oriented hierarchical data layout, and seamless layout of time-varying hierarchical data.

1. はじめに

われわれの身の回りには多くの階層型データが存在し、これを視覚化するためにさまざまな手法が提案されている。たとえば、Hyperbolic Tree[Lam96], ConeTree[Car95], Fractal Views[Koi95] は木構造を視覚化する手法である。ConeTreeに関しては、DAG 情報を視覚化した拡張手法も提案されている[Yamas02]。木構造で表現する手法のほかに、空間を分割して階層型データの構造とシェアを表示する手法[Joh91]や、3次元の入れ子構造を用いた手法[Rek93]なども提案されている。これらの視覚化技術は、データの特徴やユーザーの視覚化目的によって向き不向きがある。

筆者らは、新しい階層型データの視覚化手法として「データ宝石箱」を提案した[Ito01][Yamag02]。これは、階層構造を2次元の長方形群の入れ子構造で表して、データをできるだけ小さい画面領域に高速配置する手法である。下位階層のデータも十分に表現できるので、階層の深いデータや大規模データを視覚化するのに向いている。これを用いて、計算機のファイルシステム、会社の人事構造、ウェブサイトのサイトマップ(サイトを構成するウェブページ群の構成図)といった、巨大な階層型データを一画面に表示することができた(図1参照)。

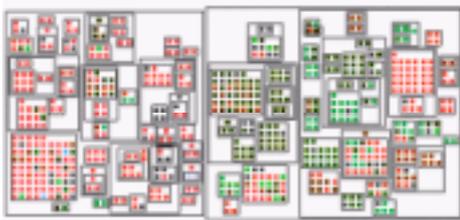


図1:「データ宝石箱」アルゴリズムを用いた配置例

本報告では、「データ宝石箱」アルゴリズムの拡張手法を提案する。本手法ではあらかじめ、階層型データを構成する各ノードの理想的な位置を、位置情報テンプレートに格納しておく。そして、テンプレートに記述された位置にできるだけ近く、しかもできるだけ小さい画面領域に、ノードを表現する長方形群を配置する。従来の「データ宝石箱」アルゴリズムはノードの配置関係をまったく制御できなかったのに対して、本手法はユーザーの意図を反映させるように配置関係を制御できるといえる。

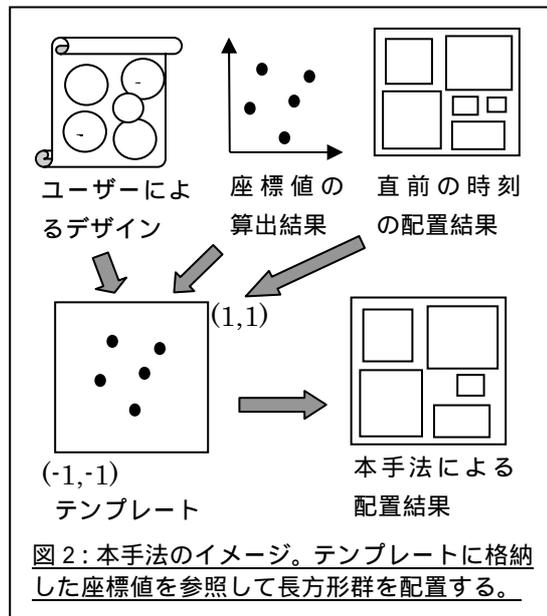
本手法によって、「データ宝石箱」の用途を以下のように拡張することができる(図2参照)。

例えば、「このデータは画面の左上に置きたい」「このデータは真ん中に置きたい」というようにデータの配置をユーザーがデザインしたいことが

あるとする。このようなときには、ユーザーのデザイン結果として得られるノードの位置をテンプレートに記述してから、本手法でノードを配置することで、ユーザーのデザイン意図をノード配置に反映することができる。

また、「新しいデータから順に左から」「アルファベット順に上から」というように、配置画面の座標軸に何らかの意味を持たせてデータを配列させたいときがあるとする。このようなときには、各ノードの座標値を算出したものをテンプレートに記述してから、本手法でノードを配置することで、何らかの意味に沿って左右または上下に並んだ配置結果を得ることができる。

また、時系列に沿って微量ずつ変化するデータを、できるだけ類似した表示結果で見比べたいときがあるとする。このようなときには、直前の時刻における配置結果をテンプレートに記述してから、本手法でノードを配置することで、シームレスな配置の変化を実現できる。



2. データ宝石箱アルゴリズム

筆者らがすでに提案している「データ宝石箱」[Ito01][Yamag02]アルゴリズムは、階層型データを2次元の長方形群の入れ子構造で表現する新しい視覚化手法である。まず、最下位階層を構成するノード群をアイコンで表現し、それを長方形で囲むことで1階層を表現する。続いて、階層を表現する大きさの異なる長方形群を隙間なく配置して、それを長方形で囲むことで上位階層を表現する。この処理を下位階層から上位階層に向けて反復することにより、大規模階層型データ全体を1画面に表示する。各長方形の面積、高さ、色などに属性を割り当てて、下位階層データのもつ統計値を多次的に表示することが可能である。

このアルゴリズムの特徴は、「長方形どうしが重

ならず、「隙間をできるだけ少なく」、大規模データを「高速に」配置する点である。1 階層を構成する長方形群が与えられたときに、まず長方形群を大きさでソートし、続いて大きい長方形から順に1つずつ配置していく。このとき、少なくとも[条件 1]、できれば[条件 2]も同時に満たす位置を探すことで長方形の位置を決定する。

[条件 1] すでに配置された長方形と、これから配置する長方形が交差しない位置。

[条件 2] これから配置する長方形の追加によって、長方形群の占有領域が拡大しない位置。

このアルゴリズムでは、長方形の位置を高速に決定するために、長方形群の中心点を連結して Delaunay 三角メッシュを生成し、これを参照しながら長方形の候補位置を探す。長方形を1個配置するたびに、その中心点を三角メッシュに追加し、「任意の三角形要素に対して生成された外接円の内部に、その三角形要素の3頂点以外の頂点を包括しない」という Delaunay 条件を満たすように三角メッシュを更新する。

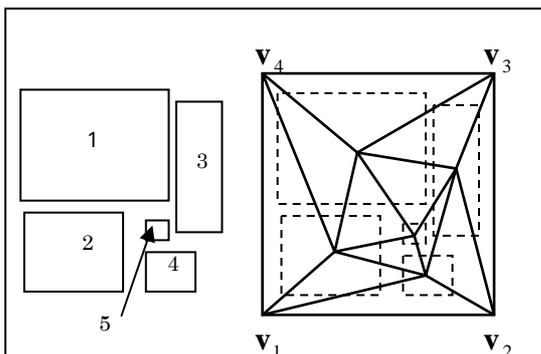


図 3: (左)「データ宝石箱」における長方形群の配置例と配置順。(右)長方形群の中心点、および占有領域の4頂点を連結して作成したDelaunay三角メッシュ。

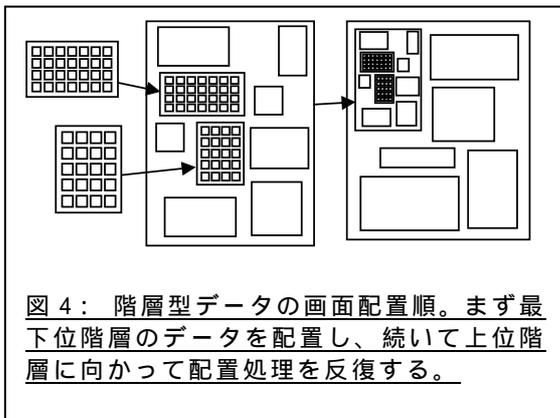


図 4: 階層型データの画面配置順。まず最下位階層のデータを配置し、続いて上位階層に向かって配置処理を反復する。

図 3(左)に、1 階層を構成する長方形群の配置例および配置順を示す。図 3(右)に、この配置を得

るために参照した Delaunay 三角メッシュを示す。図 4 に示すように、長方形群の配置処理を下位階層から上位階層に向かって繰り返すことで、階層型データ全体の配置を決定する。

3. テンプレートを用いたデータ宝石箱アルゴリズム

3.1 テンプレートの導入

1 章で述べた通り、本手法ではノードの位置情報を記録したテンプレートを導入する。テンプレートには、ノードを表現する長方形を識別するための名前と、長方形の中心点の参照座標値を格納する。参照座標値は、1 階層の占有領域を $(-1, -1) \sim (1, 1)$ の範囲に正規化した値で記録しておく。なお、このテンプレートは各階層ごとに用意するものとする。

図 2 で示した通り、ユーザーのデザインを反映させたい場合は、ユーザーがあらかじめ指定した各長方形の配置位置を記録する。また、配置画面の座標軸に何らかの意味をもたせてデータを配列させたい場合には、算出した各ノードの座標値を記録する。時系列データの場合は、直前の時刻における配置結果を記録する。

3.2 テンプレートを用いた配置

テンプレートに記述された位置にできるだけ近く、かつ 2 章で示した[条件 1][条件 2]を満たすことで「データ宝石箱」アルゴリズムの特徴を保持しながら、階層型データを表現する長方形群を画面領域に配置することを考える。そのためには長方形群を、「長方形どうしが重ならないように」「テンプレートに記録された位置にできるだけ近く」「できるだけ隙間ができないように」という 3 条件を満たすように配置する必要がある。そこで本手法では、以下の 3 基準を長方形の配置場所を決定するための判断基準とする。

[基準 1] すでに配置されている長方形群とまったく重ならない位置。

[基準 2] テンプレートに格納されている参照座標値との距離 D ができるだけ小さい位置。

[基準 3] 当該長方形の配置による占有領域の拡大量 S ができるだけ小さい位置。

処理手順は「データ宝石箱」アルゴリズムと同様に、下位階層から上位階層に向かって反復する。以下に、1 階層を構成する長方形群を、テンプレートの参照座標値を用いて配置する手順を示す。

準備として、各長方形の正規化座標系での面積を計算する。長方形の配置はすべて $(-1, -1) \sim (1, 1)$ の正規化座標系で行い、階層内の全ての長方形を配置し終えた段階で、実座標系に戻す。

1 個目の長方形の配置と初期三角メッシュの作成

まず、長方形群を配置する長方形の占有領域を作成し、占有領域の4頂点 V_1, V_2, V_3, V_4 (図3(右)参照)の座標値を $(-1, -1), (1, -1), (-1, 1), (1, 1)$ とする。次に、1階層を構成する長方形の中で面積の最も大きい長方形を、テンプレート座標値を参照して占有領域内に配置する。そして、長方形と初期領域の4頂点を結んで初期三角メッシュを作成する。

2 個目以降の長方形の配置

これ以降の配置は、テンプレートの参照座標値が最初に処理した長方形に隣接している順に行う。これは長方形の配置関係を保持するためである。

長方形の配置位置を決定するために、まずテンプレートの参照座標値を包括する三角形要素を特定する(図5(左)参照)。この三角形要素を出発点にして、隣接関係の幅優先探索によって三角形要素を1個ずつ抽出する。この三角形がテンプレート座標値の近傍 ε ($0 < \varepsilon < 1$) 内に含まれる場合、長方形の候補位置を算出すると同時に、この三角形の隣接三角形の探索を続ける。さもなければ、長方形の候補位置の算出も省略し、隣接三角形の探索も省略する。この処理の反復によって、テンプレートの参照座標値に近い三角形だけが探索される(図5(右)参照)。

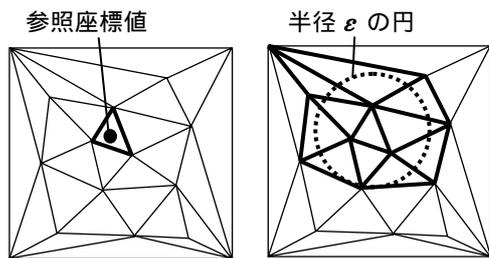


図5 : 三角メッシュとテンプレート参照座標値。
(左) テンプレート参照座標値を包括する三角形要素を太線で示したものの。(右) テンプレート参照座標値に近い太線の三角形要素だけが探索される。

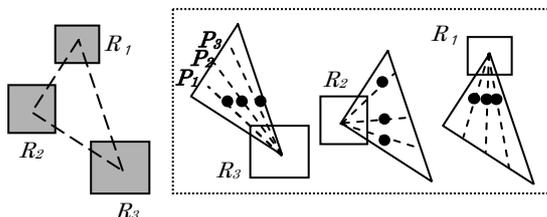


図6 : 長方形の候補位置の決定。左図のような三角形要素に対して、右図の黒丸が候補位置となる。

長方形の候補位置は、探索された三角形要素の内部に、以下のように算出する。図6(左)に示すように、三角形要素の頂点に配置された長方形を R_j とする。このとき本手法では、頂点の対辺を

n (n は任意定数) 分割する点 P_i ($i = 1, \dots, n-1$) を頂点と結び線分上で、 R_j に接する位置を、長方形の配置候補位置とする。図6(右)の黒丸で表した位置が、長方形の配置候補位置となる。

これらの候補位置が適切かどうかを判定するために、基準値として $aD+bS$ を計算する。 a, b は任意定数で、 S は長方形の面積や4辺の長さの総和などを用いる。現在の候補位置における $aD+bS$ が、記録している最小値 $(aD+bS)_{\min}$ より小さい場合にはその値を $(aD+bS)_{\min}$ に記録し、現在の候補位置を最適候補位置として記録する。これを幅優先探索によって探索された各三角形の全ての候補位置に対して判定し、最適候補位置を決定する。

以上の処理によって決定された最適候補位置に配置した長方形が、占有領域をはみ出すことがある。このとき本手法では、占有領域の各頂点 V_1, V_2, V_3, V_4 のいずれかを移動して、配置した長方形を包括するように領域を拡大する。そして、すでに配置されている長方形の中心座標値を再度正規化する。

最後に、配置した長方形の中心点を Delaunay 三角メッシュに追加し、Delaunay 条件を満たすように三角メッシュを更新する。

以上の処理を擬似コードとして表現したものを、以下に示す。

```

参照座標値を包括する三角形要素を特定する;
三角形要素を FIFO に登録する;
while (FIFO が空でない限り) {
    FIFO から三角形要素を抽出する;
    for (三角形内部の配置候補位置ごとに) {
        aD+bS 値を算出する;
        if (aD+bS < (aD+bS)min) {
            (aD+bS)min = aD + bS;
            現在の配置候補位置を記録する;
        }
    }
    for (三角形要素の隣接三角形要素ごとに) {
        if (FIFO 未登録かつ参照座標値の近傍) {
            隣接三角形要素を FIFO に登録する;
        }
    }
} // end for(..)
} // end while(..)
記録された配置候補位置に長方形を配置する;
必要に応じて占有領域の4頂点を移動する;
Delaunay 三角メッシュを更新する;

```

4. 実験

1 章で述べた通り、本手法は「ユーザーのデザ

イン意図を反映したデータ配置「座標軸に意味をもたせたデータ配置」時系列に沿って変化するデータのシームレスな配置」などへの適用が可能である。このうち本報告では、実在するウェブサイトの構成を表現した階層型データを用いて、時系列にそってウェブページが追加される様子を視覚化し、本手法の有用性を検証する。

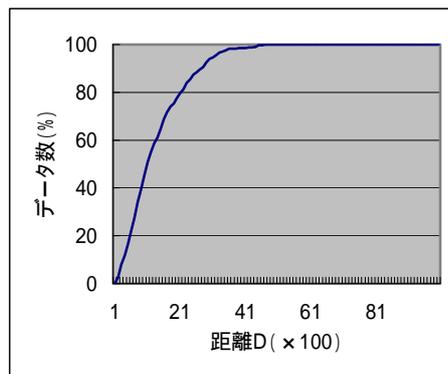
図7(左)は、実在するウェブサイトの構成を表現した階層型データを、「データ宝石箱」アルゴリズムを用いて表示した結果である。このウェブサイトが、次の時刻で1つのディレクトリに2ページを追加していたとする。このデータを「データ宝石箱」アルゴリズムを用いて表示すると、図7(中)のような配置結果が得られた。図7(左)と比べると、データの増加が少ないのにも関わらず、視覚結果が大きく異なっている。この結果は、「データ宝石箱」アルゴリズムが、類似するデータにおいても非常に異なる配置結果を生じることがある、という問題があることを示している。

ここで図7(左)の配置結果をテンプレートとし、本手法を用いて、図7(中)と同じデータを再配置した。[基準3]より[基準2]に重みをつけるため、判定基準である $aD+bS$ の定数 a, b は $a=5, b=1$ とした。本手法による配置結果を図7(右)に示す。定性的に評価すると、図7(左)に近い配置結果が得られることから、本手法は、階層型データの時系列変化に対してシームレスな視覚的結果を与えているといえる。

次に、本手法を用いた配置結果がテンプレートの参照座標値にどの程度近く配置しているかを調べるために、テンプレートの参照座標値との距離 D を測定した。図7では定性的に評価できるようにデータ数が少ないもので実験を行ったが、ここでは定量的に評価するためにデータ数が十分多いデータを用いた。図8にその配置例を示す。図8(左)は「データ宝石箱」アルゴリズムを用いた配置結果である。図8(右)は、図8(左)に示した配置結果の座標値をテンプレートに記述し、図8(左)と同じデータに対して本アルゴリズムを用いて再配置した結果である。定量的評価を行うためにテンプレートと同じデータを使用した。全ての長方形に対し距離 D を計測し、その相対累積度数分布をグラフ1に示した。ここで D の平均値は0.15であった。またグラフ1を見ると、 D 値が0から0.24の範囲でデータ全体のほぼ80%を占めていることがわかった。このことから本アルゴリズムは、テンプレートの参照座標値に十分近い位置に長方形群を配置しているといえる。

最後に、本手法によってどのような形状の長方形が生成され、どのくらい大きな占有領域にデータが配置されたかについて、定量的に評価する。表1は、参考文献[Yamag02][Bed01]の比較項目であったアスペクト比および空領域比を、図8の配置結果に対して測定した。各項目の説明は参考文献を参照されたい。表1から空間占有率もアスペクト比も、テンプレートとして用いた配置結果と、

本手法による配置結果とで、ほぼ等値な配置が得られたことがわかる。このことから本手法は、テンプレートの参照座標値に近い位置にデータを配置できただけでなく、「データ宝石箱」アルゴリズムの特徴を継承するようなデータ配置結果を得ることもできたといえる。



グラフ1：配置誤差 D の累積度数分布

表1：定量的評価。(左)テンプレートとして用いた配置結果。(右)本手法を用いて再配置した結果。

	図8(左)	図8(右)
ノードの数	1617	1617
空領域比	0.5320	0.5262
アスペクト比	1.1745	1.1743

5. おわりに

本報告では、テンプレートに格納した位置にできるだけ近く、できるだけ小さい長方形領域に、かつ互いに重ならないように、という3条件を満たすように長方形群を配置するアルゴリズムを提案した。また、このアルゴリズムを用いて階層型データを表現する手法を提案した。

本アルゴリズムでは、すでに配置された長方形群の中心点を連結する三角メッシュを生成し、これを参照して長方形の候補位置を算出する。このとき、テンプレートに格納された参照座標値に近い三角形要素だけを参照することで、計算を高速化する。また、すでに配置された長方形に重ならない位置で、かつテンプレート参照座標値との距離および画面領域の拡大量から得られる評価値が最小となる候補位置を選ぶことで、上記の3条件を満たす配置を実現する。

本アルゴリズムにより、「類似した階層構造データに対し、全く異なる配置結果が得られる場合がある」という「データ宝石箱」アルゴリズムの問題点を解決できたといえる。

本手法を用いることで、時間変化によって変化

するデータの視覚化だけでなく、ユーザーのデザインを反映したデータ視覚化、配置軸に意味をもたせたデータ視覚化、などの目的にも「データ宝石箱」を拡張できると考えられる。これらのデータに対する実験を今後の課題としたい。

参考文献

[Ito01] 伊藤, 梶永, 池端, データ宝石箱: 大規模階層型データのグラフィックスショーケース, 情報処理学会グラフィックスとCAD研究報告, 2001-CG-104, 2001.
 [Yamag02] 山口, 伊藤, 池端, 梶永, サイトマップ表示とアクセス統計表示の連携によるウェブサイト視覚化ツール, 情報処理学会論文誌投稿中.
 [Yamas02] 山下, 藤代, 高橋, 堀井, 拡張ConeTrees技法によるDAG情報の可視化, Visual Computing グラフィックスとCAD合同シンポジウム 2002, pp.1-6, 2002.
 [Ber98] Berderson B.B., et al., A Zooming Web Browser, *Human Factors in Web Development*, pp. 255-266, 1998.
 [Car95] Carriere J., et al., Research Report: Interacting with Huge Hierarchies Beyond Cone Trees, *IEEE Information Visualization '95*, pp. 74-81, 1995.
 [Joh91] Johnson B., et al., Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, *IEEE Visualization '91*, pp.

275-282, 1991.

[Koi95] Koike H., Fractal Views: A Fractal-Based Method for Controlling Information Display, *ACM Trans. on Information Systems*, 13, 3, pp. 305-323,
 [Lam96] Lamping J., Rao R., The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, *J. Visual Languages and Computing*, 7, 1, 33-55, 1996.
 [Rek93] The Information Cube: Using Transparency in 3D Information Visualization, Third Annual Workshop on Information Technologies & Systems, pp. 125-132, 1993.
 [Bed01] Bederson B., PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps, *UIST 2001*, pp. 71-80, 2001.

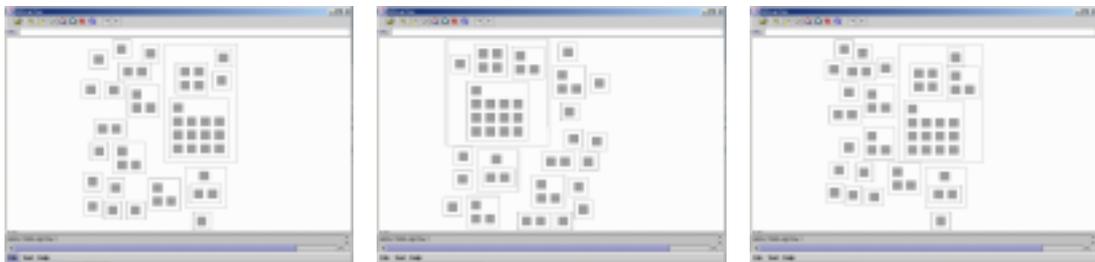


図 7: 時系列にそって変化するウェブサイトの構造を階層型データで表現して視覚化した結果。(左) 直前の配置結果。(中)「データ宝石箱」アルゴリズムを用いて配置した結果。(右) 直前の配置結果をテンプレートにして本手法を適用した結果。

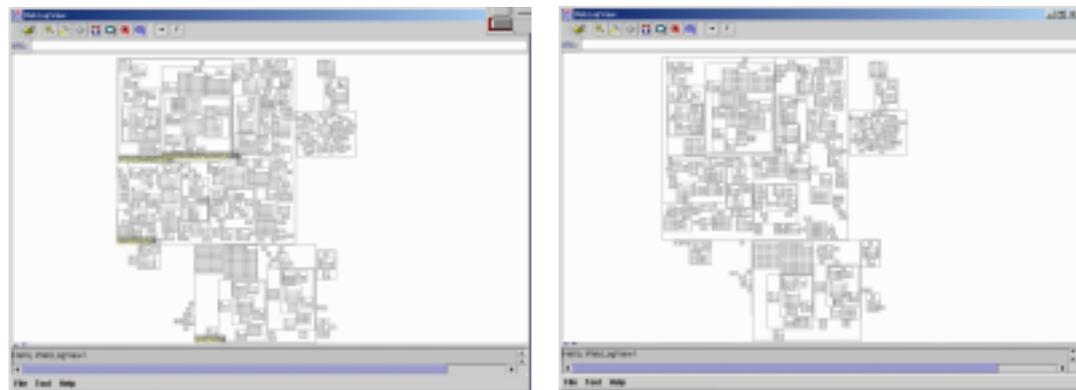


図 8: ウェブサイトの構造を階層型データで表現して視覚化した結果。(左)「データ宝石箱」アルゴリズムを用いて配置した結果。(右) 左図をテンプレートにして、本アルゴリズムで配置した結果。