長方形の入れ子構造を用いた階層型データ視覚化手法の拡張

山口裕美† 伊藤貴之†

筆者らは，大規模階層型データを長方形の入れ子構造で表現する視覚化手法「データ宝石箱」を提案している．この手法は階層型データ全体を 1 画面に配置して表示するので，階層型データの全貌を一目で眺観することができる．しかしながらこの手法は，データの意味やユーザの意図に対して一意な画面配置を得ることができなかったので，たとえば類似したデータに対して非常に異なる画面配置結果を生じる，というような問題点があった．本論文では，階層型データを構成するノードの理想位置を記述したテンプレートを生成し，これを参照しながら階層型データを 1 画面に表示する手法を提案する．本論文ではこの提案手法を「データ宝石箱 II」と呼ぶ．本手法では,「ノードどうしが重ならないように」「画面の占有領域ができるだけ小さく，かつ正方形に近い形状になるように」「高速に」という「データ宝石箱」の要求に加えて,「テンプレートに記述された位置にできるだけ近い位置に」という新しい要求を加えて，階層型データを構成するノードを画面に配置する．

An Extension of Hierarchical Data Visualization Technique Using Nested Rectangles

Yumi Yamaguchi† and Takayuki Itoh†

We have proposed “Data Jewelry Box” as a hierarchical data visualization technique which represents the data by nested rectangles. The technique provides good overviews of the data, because it represents the whole data in one display space. However, it has a problem that it does not trivially calculate display layout results according to semantics of the data or user’s intension. This paper proposed the extended technique, “Data Jewelry Box II”, which places hierarchical data onto display spaces. The extended technique refers templates that describe the ideal position of nodes while it places the nodes onto the display space. Requirement of original “Data Jewelry Box” was “no overlap among nodes”, “smaller and regular display area”, and “quickly”. In addition to the above requiments, the extended technique satisfies one more requirement, “close to the ideal positions described in templates”.

1. はじめに

我々の身の周りには多くの階層型データが存在している．これを視覚化するために，筆者らは「データ宝石箱」という視覚化手法を提案した[11)]．この手法は，階層型データの葉ノードを長方形のアイコンで，枝ノードを長方形の枠で表現し，階層構造を 2 次元の長方形群の入れ子構造で表現したものである（**図 1** 参照）．

本手法は，階層型データ中の葉ノードと枝ノードの親子関係よりも，階層型データ全体に分布する葉ノード群の一覧表現に重点をおいた視覚化手法であるということができる．筆者らはこの手法を適用して，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）といった大規模な階層型データの全貌を，1 画面に表示することを試みている．また，すべての葉ノードを 1 画面に表示するという方針が，ウェブサイトのアクセス傾向の視覚化において有効であったことを，参考文献 11) に示している．

階層型データ視覚化手法において重要な技術は，階層型データを構成するノード群を適切に画面配置する技術である．ノード群の画面配置に関して筆者らは，以下のような要求を満たす視覚化手法の開発を目標としている．

要求 1 すべてのノードを同時に表示するために，互いに重ならないようにノードを配置する．

要求 2 画面空間を節約するために，ノード群の配置領域となる長方形領域が，できるだけ小さい面積になるように，かつできるだけ正方形に近い形状になるようにノードを配置する．

要求 3 データの意味やユーザの意図を反映した一意

† 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research, Tokyo Research Laboratory

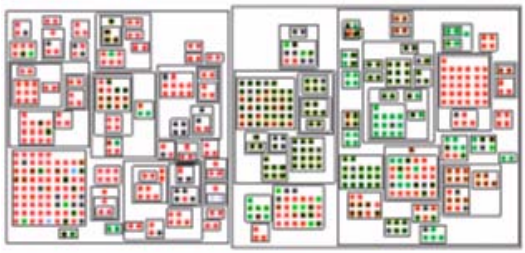

図 1 「データ宝石箱」アルゴリズムを用いた配置例
Fig. 1 Example of hierarchical data visualization using "Data Jewely Box".

な画面配置を実現する．

要求 4 できるだけ高速に配置する．

これらの要求のうち，筆者らが提案している「データ宝石箱」は,［要求 3］を満たしていなかった．そのため，たとえば，類似しているデータに対して，まったく異なる画面配置結果をもたらす場合があった．

本論文は，この問題を解決する拡張手法「データ宝石箱 II」[12]を提案する.「データ宝石箱 II」では，階層型データを構成する各ノードに対して，理想的な画面位置を記録した理想座標値をあらかじめ用意する．そしてその理想座標値を参照しながら，階層型データを画面配置する．本論文では，この理想座標値を記録したデータを「テンプレート」と呼ぶ．

「データ宝石箱 II」は，以下のような用途（**図 2** 参照）において，上記の［要求 1］〜［要求 4］をすべて満たす視覚化手法として有効であると考えられる．

ユーザのデザイン意図を反映した画面配置：たとえば,「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がデザインしたいことがあるとする．このとき，ユーザ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録し，これを参照しながらノードを画面配置することで，ユーザのデザイン意図を視覚化結果に反映することができる．

座標軸に意味を持たせた画面配置：「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，各ノードに対して算出した座標値をテンプレートに記録し，これを参照しながらノードを画面配置することで，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．

時系列に沿って微量ずつ変化するデータのシームレスな視覚化：時系列に沿ったデータに本手法を適用する際には，直前の時刻における画面配置結果をテンプレートに記録する．これを参照しながらノードを画面配置することで，シームレスに変化する画面配置を実現できる．

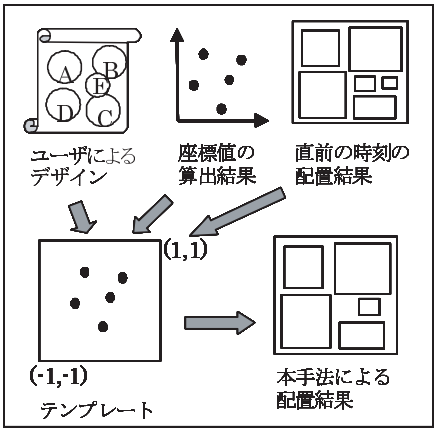

図 2 本手法のコンセプト．テンプレートに記録した理想座標値を参照して長方形群を配置する
Fig. 2 Concept of "Data Jewelry Box II".

本論文では，次章で関連研究をいくつか紹介したあと，3 章で著者らがすでに提案した「データ宝石箱」を紹介し，4 章で本論文の提案手法である「データ宝石箱 II」を説明する．特に，提案手法で導入するテンプレートの概念と，これを用いた新しい長方形画面配置アルゴリズムについて説明する．続いて 5 章で，提案手法をいくつかのデータに適用した例を示した後，6 章でまとめを述べる．

2. 関 連 研 究

本章では，階層型データの視覚化に関する従来手法，およびその他の画面配置手法について述べる．

2.1 木構造表示型の階層型データ視覚化手法

階層型データの視覚化手法で最も一般的な表示方法は，木構造を表示する方法である．大規模階層型データの対話的な探索に向いた手法として，Hyperbolic Tree[7]，Cone Tree[2]，Fractal Views[6]などの各手法が提案されている．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[14]．

これらの視覚化手法は，階層型データを構成する葉ノードをすべて 1 画面に表示するというよりも，上位階層から下位階層に向けてデータを探索する際の，ユーザインタフェースとして利用されることが多い．その観点から，筆者らの提案手法とは目的が異なるといえる．

2.2 空間分割型の階層型データ視覚化手法

与えられた画面領域を分割して階層型データを表示

する手法として，入れ子状の棒グラフで階層型データを表示する TreeMaps[5]や，階層構造に従って円グラフを内側から外側に積み上げる手法[3]などがあげられる．これらの手法は，階層型データ全体を 1 つの画面領域に表現する，という観点から筆者らの提案手法に類似している．

TreeMaps は，下位階層が非常に細長くなって視覚的に認識できなくなるケースが多いこと，また下位階層データをアイコンで表現することが難しいことなどが問題となっていた．これらの問題点に対する改善手法[1),10)]が発表されているが，依然として 1 章で述べた要求をすべて満たしているとはいえない．たとえば Ordered Treemap[10]では，葉ノードを同一形状で表現することが困難である．また時系列データに適用したときに，一部のノードが画面上で非常に大きく移動することが指摘されている．また Bubblemap[1]では，1 階層を表現する長方形領域が非常に細長くなることがある．

2.3 3 次元の入れ子構造を用いた階層型データ視覚化手法

筆者らの提案手法は，長方形の入れ子によって 2 次元的に階層構造を表現しているが，これに類似した発想で，半透明な直方体の入れ子によって 3 次元的に階層構造を表現する Information Cube[9]が知られている．この手法には，半透明表示のできるグラフィックス環境が必要であること，ユーザ側に 3 次元 CG の操作スキルを要すること，階層構造が深いときに透明度の調節が難しい，などの制限がある．

2.4 階層型データ以外のデータに対する画面配置手法

階層型データ以外のデータ視覚化においても，データを構成するノード群を有効に画面配置する問題は，重要な問題として議論されている．

画面領域を有効に使うという観点で有名な手法として，Design Gallery[8]が知られている．Design Gallery では，多次元変数を与えられたノード群に対して，画面上に最も分散して配置されるように，2 次元の座標軸を自動算出する．しかしこの手法でも，画面上でのノードの重なりを避けることは難しい．

また，画面領域を有効に使うために，力学モデルを用いてノード間の距離を適正に保つ手法が，グラフデータ視覚化[4]などの分野で知られている．しかしこの手法は，計算時間がかかることが多い．

3. データ宝石箱

本章では，筆者らの従来手法である「データ宝石箱」[11]について述べる．この手法は，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として提案された．

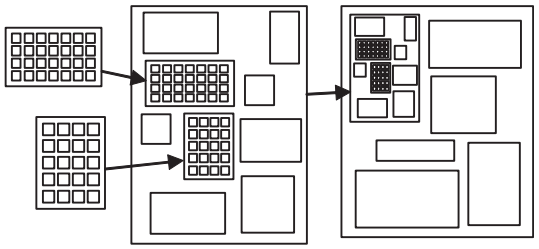

図 3 階層型データの画面配置順．まず最下位階層のデータを配置し，続いて上位階層に向かって配置処理を反復する
Fig. 3 Order of placing all nodes in hierarchical data.

3.1 階層型データ全体の配置

「データ宝石箱」は，データを構成する葉ノードをアイコンで，枝ノードを長方形の枠で表現し，階層型データ全体を 1 画面に配置する．配置方法の概要を**図 3** に示す．「データ宝石箱」ではまず，最下位階層を構成する葉ノードを格子状に配置する．これを長方形で囲むことで，1 つの枝ノードを表現する．続いて，その上位階層を構成するノードを敷きつめて，それを長方形で囲んで 1 つの枝ノードを表現する．同様な処理を，下位階層から上位階層に向かって繰り返すことにより，階層型データ全体を画面配置する．

以下，1 階層を構成する葉ノードおよび枝ノードを表現する長方形群の画面配置方法について説明する．

3.2 1 階層内での長方形の配置位置決定アルゴリズム

「データ宝石箱」では，1 章であげた要求のうち［要求 1］，［要求 2］を，以下の要件と置き換えて長方形を配置する．

要件 1 これから配置する長方形を，すでに配置された長方形にまったく重ならないように配置する．

要件 2 これから配置する長方形を，長方形群の占有領域が拡大しない位置に配置する．もしそのような位置が見つからない場合には，拡大量が小さく，かつ占有領域が正方形に近くなる位置に配置する．

「データ宝石箱」では，上記の要件を満たす位置を高速に算出するために，すでに配置されている長方形群の中心点を連結する Delaunay 三角メッシュ（**図 4** (左) 参照）を生成し，これを参照しながら長方形の候補位置を求める．詳細を以下で述べる．

3.2.1 三角メッシュの生成と 1 個目の長方形の配置

まず最初に，1 階層を構成する長方形群の中から，面積が最大である長方形を選ぶ．その長方形を画面空間の中心に配置し，さらにその長方形を囲む長方形領

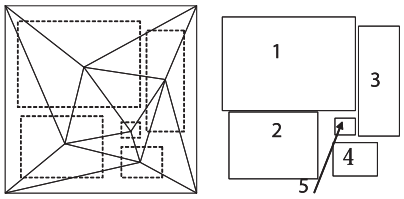

図 4 (左) 長方形群の中心点，および占有領域の 4 頂点を連結して作成した Delaunay 三角メッシュ．(右)「データ宝石箱」における長方形群の配置例と配置順

Fig. 4 (left) Delaunay triangular mesh connecting centers of rectangles and four couners. (right) Order of placing rectangles.

域を定義する．本論文では，この領域を占有領域と呼ぶ．そして占有領域を構成する 4 頂点と，配置された長方形の中心点，の合計 5 頂点を連結する初期三角メッシュを生成する．

3.2.2 2 個目以降の長方形の配置手順

長方形の配置順

「データ宝石箱」では図 4 (右) に示すように，長方形を面積の大きい順に配置する．この配置順は，まず大きい長方形を配置し，続いてその隙間に小さい長方形を配置したほうが，占有面積が小さくなる可能性が高い，という経験則に基づいている．

長方形の候補位置算出の反復処理

「データ宝石箱」では，三角メッシュを構成するメッシュ辺の上に長方形の候補位置を算出する処理を反復する．メッシュ辺の処理順，および候補位置の算出方法については，参考文献 11) で提案している．

長方形の配置位置の決定

上述の方法で算出された候補位置に長方形の配置を試みて，その候補位置が［要件 1］,［要件 2］の両方を満たすようであれば，候補位置算出の反復処理を終了し，その位置に長方形を配置する．その候補位置が［要件 1］のみを満たすようであれば，その位置に長方形を配置したことによる占有領域の拡大量を評価する．すべての候補位置に対して長方形の配置を試みて，もし［要件 1］,［要件 2］を満たす候補位置がまったく見つからない場合には，占有領域の拡大量の評価が最も良かった候補位置に長方形を配置する．占有領域の拡大量の評価については，参考文献 11) で提案している．

三角メッシュの更新

「データ宝石箱」では，長方形を 1 個配置するたびに，その長方形の中心点を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．ここで Delaunay 条件とは,「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含しない」という条件である．Delaunay 条件を適用している理由は，三角メッシュ中に細長く歪んだ三角形要素が少ないほど，適切に候補位置を算出できる，という経験則によるものである．

なお，いま配置した長方形が占有領域をはみ出すときには，占有領域の各頂点のいずれかを移動して，いま配置した長方形を包含するように占有領域を拡大してから，三角メッシュの更新を行う．

4. 画面配置情報を記述したテンプレートを用いた階層型データの配置アルゴリズム

本章では，本論文で提案する「データ宝石箱 II」について詳しく述べる．

4.1 テンプレートの導入

「データ宝石箱 II」では，ノードの画面上の理想座標値を記録した,「テンプレート」というデータを用いる．テンプレートの具体的な実用方法は，図 2 で示したとおりである．なお，このテンプレートは階層型データの各階層ごとに用意するものとする．

テンプレートには，ノードを識別するための名前と，ノードを表現する長方形の中心点の理想座標値を格納する．ここで参照座標値は，1 階層の占有領域を $(-1,-1)$〜$(1,1)$ の範囲に正規化した値で記録する．このため「データ宝石箱 II」では，まずノードを表現する長方形を $(-1,-1)$〜$(1,1)$ に正規化された座標値で配置し，続いてノードの座標値を実座標系に変換する．

階層型データの 1 階層の中に，テンプレートに理想座標値を記述しているノードと記述していないノードが混在していてもよい．この場合，まずテンプレートに理想座標値が記述されているノードを正規化された座標値で画面配置して，それらを実座標に変換する．続いて，テンプレートに理想座標値を記録されていないノードを，3 章で紹介した「データ宝石箱」により画面配置する．

4.2 階層型データ全体の配置

「データ宝石箱 II」は「データ宝石箱」と同様，図 3 に示すように，下位階層から上位階層に向けてノードの画面配置処理を反復する．

4.3 テンプレートを用いた 1 階層内での長方形の配置アルゴリズム

「データ宝石箱 II」では，1 章であげたような要求をすべて満たすために，3 章で述べた「データ宝石箱」の要件だけでなく，以下の要件も同時に満たす位置に長方形を配置する．

要件 3 テンプレートに記録された理想座標値にでき

るだけ近い位置に，長方形を配置する．

以下，テンプレートに記録された理想座標値を参照しながら，画面上に長方形を配置する新しいアルゴリズムを提案する．

4.3.1 三角メッシュの作成と 1 個目の長方形の配置

まず最初に，これから長方形群を配置していく占有領域を作成する．占有領域の 4 頂点の座標値をそれぞれ $(-1,-1)$，$(1,-1)$，$(-1,1)$，$(1,1)$ とする．

「データ宝石箱 II」ではテンプレートに記録されている長方形の中で，面積が最大であるものを最初に画面配置する．そして「データ宝石箱」と同様，配置された長方形の中心点と，それを包含する占有領域の 4 頂点，の合計 5 点を結んで初期三角メッシュを作成する．

4.3.2 2 個目以降の長方形の配置

長方形の配置順の決定

2 個目以降の長方形の配置順は，テンプレートに記録された長方形の座標値のうち，最初に配置した長方形に距離が近い順に配置していく．これは，長方形の画面上での隣接関係を保持するためである．

長方形の候補位置算出の反復処理

「データ宝石箱」では三角メッシュのメッシュ辺上に候補位置を算出したが,「データ宝石箱 II」では，より多くの要件を同時に満たす位置を探し出すために，より細かく候補位置を算出する必要がある．そこで「データ宝石箱 II」では，三角メッシュの三角形内部に候補位置を算出する．

「データ宝石箱 II」ではまず，テンプレートに記録されている長方形の理想座標値を包含する三角形要素を特定する（**図 5** (左) 参照）．この三角形を出発点にして，隣接関係の幅優先探索によって三角形を 1 個ずつ抽出する．この三角形が理想座標値の近傍$\epsilon(0<\epsilon<1)$ 内に含まれる場合，この三角形の内部に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を省略し，隣接三角形の探索も省略する．この処理の反復によって，テンプレートの理想座標値に近い三角形だけが探索される（図 5 (右) 参照）．

続いて探索された三角形の内部に，長方形の候補位置を算出する．三角形要素を構成する頂点 $j(j=1,2,3)$ 上にすでに長方形 R_j が配置されている場合には，隣接長方形間に不必要な隙間をつくりたくないので，これから配置する長方形が R_j に接するように候補位置を算出する．具体的には**図 6** に示すとおり，頂点 j に対して角の n 等分線を引き（n は任意定数），その線上で長方形 R_j に接する位置を新しい長方形の候補位置とする．もし，頂点 j が占有領域の頂点である場合は，頂点 j に対する候補位置の算出を省略する．以上の処理を，三角形要素の他の 2 頂点に対しても同様に行う．

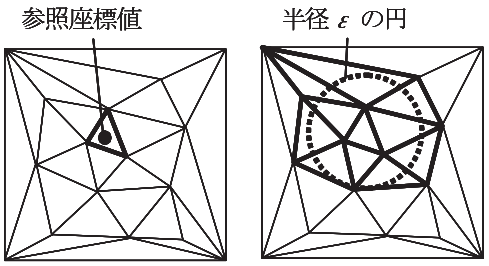

図 5 三角メッシュと理想座標値．(左) テンプレート理想座標値を包含する三角形要素を太線で示したもの．(右) テンプレート理想座標値に近い太線の三角形要素だけが探索される

Fig. 5 Triangular mesh and an ideal position described in a template. (left) A triangle enclosing the ideal position. (right) Triangles near the ideal position.

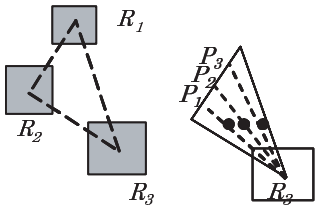

図 6 長方形の候補位置の決定．左図のような三角形要素に対して，右図の黒丸が候補位置となる

Fig. 6 Candidate positions in one triangle.

長方形の配置位置の決定

続いて，各々の候補位置について長方形の配置を試み，すでに配置されている長方形との交差を判定する．もし長方形との交差があるなら，この候補位置は［要件 1］を満たさないので，以後の処理を省略する．

［要件 1］を満たす複数の候補位置の中から,［要件 2］,［要件 3］を両方満たす位置を決定するために,「データ宝石箱 II」では各々の候補位置における判定基準値 $aD+bS$ を算出し，この値が最小となる位置を長方形の配置座標値に決定する．ここで D は，テンプレートに記録された理想座標値と，候補位置の正規化された座標値とのユークリッド距離をさす．S は，新しく長方形を配置した後の占有領域の評価値を表す．筆者らの実装では，長方形を配置する前後の占有領域について，面積の増加率と，縦横比の比を算出し，この和として占有領域の評価値としている．a，b は任意定数で，それぞれ D と S に重みをつける．［要件 3］を重視したいときは a の値を大きくし，逆に［要件 2］を重視したいときは b の値を大きくする．

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

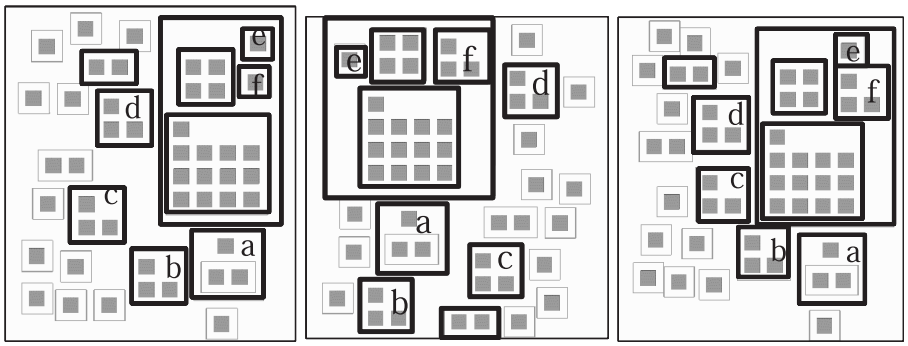

図 7 時系列に沿って変化するウェブサイトの構造を階層型データで表現して視覚化した結果. (左) 直前の配置結果. (中)「データ宝石箱」を用いて配置した結果. (右) 直前の配置結果をテンプレートに記録して「データ宝石箱 II」を適用した結果

Fig. 7 Experiment1: Layout of time-varying data. (left) Layout of previous data. (center) Layout using "Data Jewelry Box". (right) Layout using "Data Jewelry Box II".

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すべての候補位置に対して判定し，最適な候補位置を決定する．

三角メッシュの更新

「データ宝石箱 II」では「データ宝石箱」と同様に，いま配置した長方形の中心点を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．

なお，いま配置した長方形が占有領域をはみ出すときには，占有領域の各頂点のいずれかを移動して，いま配置した長方形を包含するように占有領域を拡大する．そして，すでに配置されている長方形群の中心座標値を再度正規化した後，三角メッシュの更新を行う．

5. 実　　験

1 章で述べたとおり,「データ宝石箱 II」は,「ユーザのデザイン意図を反映したデータ配置」,「座標軸に意味を持たせたデータ配置」,「時系列に沿って変化するデータのシームレスな配置」などを可能にする．このうち本論文では,「座標軸に意味を持たせたデータ配置」と「時系列に沿って変化するデータのシームレスな配置」の実現例を示す．階層型データの例として本論文では，実在するウェブサイトを構成するウェブページを，ディレクトリ階層に従って階層型データにしたもの[11]を適用した．

5.1 時系列に沿って変化するデータの画面配置

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」を用いて表示した結果である．このウェブサイト中の 1 つのディレクトリに，次の時刻で 2 ページが追加されたとする．新しく 2 ページ追加されたウェブサイトを表現する階層型データを，従来の「データ宝石箱」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，微量なノードの増加にもかかわらず，画面配置結果が大きく異なっている．

ここで，図 7 (左) の配置結果をテンプレートに記録し,「データ宝石箱 II」を用いて，図 7 (中) と同じデータを再配置した.［要件 2］より［要件 3］に重みをつけるため，判定基準である $aD+bS$ の定数は，$a = 5$, $b = 1$ とした.「データ宝石箱 II」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．

5.2 座標軸に意味を持たせた画面配置

本節では，ウェブサイトを構成するファイルやディレクトリを，ファイル名またはディレクトリ名の辞書順で配置画面の x 軸に沿って配置し，ファイルまたはディレクトリのアクセス数の昇順で配置画面の y 軸に沿って配置した結果を示す．

図 8 (中) は，判定基準 $aD + bS$ の定数を $a = 1$, $b = 1$ としてデータを画面配置した結果である．この図から，左から順にファイルやディレクトリがアルファベット順に並んでいることが分かる．図 8 (右) では，辞書順やアクセス数に忠実に配置するために,［要件 3］に重みをつけ，判定基準 $aD+bS$ の定数を $a = 5$, $b = 1$ とした．図 8 (右) の配置結果と，図 8 (中) の配置結果を比較すると，図 8 (右) の方が配置順を忠実に反映した配置を実現している．しかし，各座標軸に忠実に配置している反面，空領域が多く，そのために全体の占有領域が大きいことが分かる．一方，図 8 (左)

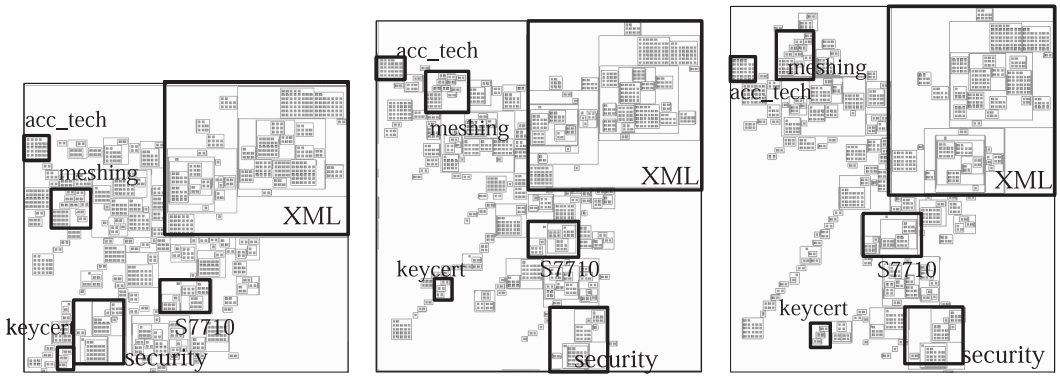

図 8 座標軸に沿って配置した結果（x 軸：アルファベット順．y 軸：アクセス数）．(左) 占有面積重視．(中) 標準．(右) 配置重視

Fig. 8 Placement of a rectangle and local modification of a triangle mesh.

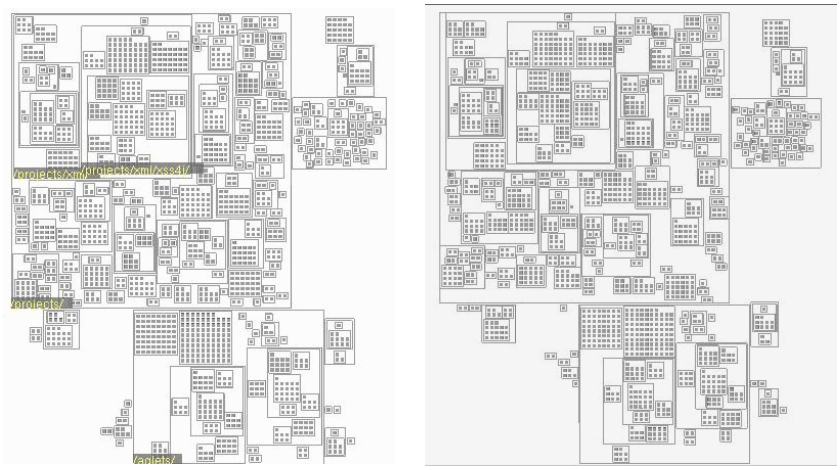

図 9 ウェブサイトの構造を階層型データで表現して視覚化した結果．(左)「データ宝石箱」アルゴリズムを用いて配置した結果．(右) 左図をテンプレートにして,「データ宝石箱 II」で配置した結果

Fig. 9 Experiment2: Layout of a web site data. (left) using “Data Jewelry Box” algorithm. (right) using “Data Jewelry Box II” referring templates which describes the left layout.

では，全体の占有領域をできるだけ小さくするために判定基準の定数を $a = 1$，$b = 5$ とした．すると，図 8 (中) や図 8 (右) と比較してデータが中央に充填して配置されており，全体の占有領域が小さいことが分かる．

5.3 定量的評価

本節では,「データ宝石箱 II」を用いた配置結果が，テンプレートに記録された理想座標値にどの程度近いかを調べるために，正規化された座標系における理想座標値との距離 D を測定した．**図 9** に，葉ノード数 5,684 個によって構成される階層型データの配置例を示す．図 9 (左) は「データ宝石箱」を用いた配置結果である．図 9 (右) は，図 9 (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 9 (左) と同じデータに対して「データ宝石箱 II」を用いて再配置した結果である．

ここで，図 9 (右) に示した結果について，すべてのノードに対し距離 D を計測し，その相対累積度数分布を**図 10** に示した．ここで D の平均値は 0.15 であった．また図 10 を見ると，D 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分かった．なお D の最大値は，正規化された座標系でノードが $(-1, -1)$ から $(1, 1)$ に移動したときであり，このとき $D = 2.828$ である．大半のノードにおいて，D の値が最大値と比べて非常に小さいことから,「データ宝石箱 II」はテンプレートの理想座標値に十分近い位置

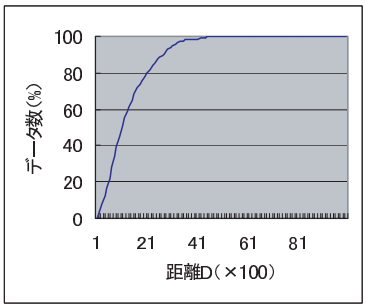

図 10 配置誤差 D の累積度数分布グラフ
Fig. 10 Frequency distribution graph of distance between ideal and actual position.

表 1 図 9 の配置結果に対する定量的評価
Table 1 Numerical evaluation of layout results shown in Fig. 9.

	図 9 (左)	図 9 (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

に長方形群を配置しているといえる．

続いて,「データ宝石箱 II」によって生成される長方形の形状および大きさを定量的評価する．**表 1** は，図 9 の配置結果に対して，参考文献 1)，11) の比較項目であったアスペクト比および空領域比と，処理時間を測定した結果である．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，テンプレートに記述した配置結果と,「データ宝石箱 II」による配置結果とで，ほぼ等値な配置が得られたことが分かる．このことから「データ宝石箱 II」は，テンプレートの理想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」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．

最後に,「データ宝石箱 II」を用いた際の処理時間と，配置結果の理想座標値との距離 D の平均値を測定し，その相関性を分析する．**表 2** は，葉ノード数 4,080 個で構成される階層型データに対して，4.3.2 項で示した ϵ 値を調節し，また候補位置どうしの距離がおおむね l になるように n 値を可変にして,「データ宝石箱 II」を用いて画面配置を算出した結果である．

この結果より,「データ宝石箱 II」の処理時間は,「データ宝石箱」よりも大きくなる傾向にあることが分かる．これは「データ宝石箱 II」が,「データ宝石箱」に比べて多くの候補位置を処理していることに起因する．また，ϵ 値および l 値を調節すると，処理時間および配置結果が大きく変化することが分かる．特に，距離 D の平均値が小さい方が，テンプレートの理想座標値に近いことから，好ましい位置にデータを配置していることが分かる．これらのことから,「データ宝石箱 II」を有効に実用するためには，計算時間と配置結果のバランスをとったパラメータ値を見つける必要があることが分かる．有効なパラメータの発見方法，また計算時間と配置結果を両立するためのアルゴリズム改善などが，今後の課題としてあげられる．

表 2 処理時間と配置結果の相関性
Table 2 Trade off between computation time and quality of layout.

アルゴリズム	ϵ	l	処理時間（秒）	D 平均値
データ宝石箱	-	-	0.480	-
データ宝石箱 II	0.5	0.1	11.907	0.170
データ宝石箱 II	0.5	0.3	4.626	0.178
データ宝石箱 II	0.1	0.1	11.083	0.171
データ宝石箱 II	0.1	0.3	4.236	0.179

6. おわりに

本論文では，1 章で掲げた 4 つの要求をすべて満たすように階層型データを画面配置する視覚化手法「データ宝石箱 II」を提案した．

提案手法では，階層型データ中の各ノードに対して，画面上での理想座標値をテンプレートに記録し，それを参照しながらノードを画面配置する，という新しい概念を導入している．

また提案手法では，3 章および 4 章で示した 3 つの要件をすべて満たす新しい長方形画面配置アルゴリズムを導入している．このアルゴリズムでは，すでに配置されている長方形群の中心点を連結する三角メッシュを生成し，この三角メッシュを参照して長方形の候補位置を算出する．そしてその候補位置の中で，すでに配置された長方形に重ならず，かつ理想座標値との距離と占有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に長方形を配置することで，上記の要件を満たす配置を実現した．このとき，テンプレートに記録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．

紙面の都合で紹介できなかったが，筆者らは「データ宝石箱 II」を，分散計算環境上のプロセス分布を監視するプロトタイプシステムに適用している[13]．

今後の課題として，5.3 節で記述したとおり，処理時間と配置結果の両立を目指すためのアルゴリズム改善があげられる．

参考文献

1) Berderson, B.B.: PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and

Bubblemaps, *UIST 2001*, pp.71–80 (2001).
2) Carriere, J., et al.: Research Paper: Interacting with Huge Hierarchies beyond Cone Trees, *IEEE Information Visualization 95*, pp.74–81 (1995).
3) Chuah, M.: Dynamic Aggregation with Circular Visual Designs, *IEEE Information Visualization '98*, pp.35–43 (1998).
4) Herman, I., Melancon, G. and Marshall, M.S.: Graph Visualization and Navigation in Information Visualization: A Survey, *IEEE Trans. Visualization and Computer Graphics*, Vol.6, No.1, pp.24–43 (2000).
5) Johnson, B., et al.: Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, *IEEE Visualization '91*, pp.275–282 (1991).
6) Koike, H.: Fractal Views: A Fractal-Based Method for Controlling Information Display, *ACM Trans. Inf. Syst.*, Vol.13, No.3, pp.305–323 (1995).
7) Lamping, J. and Rao, R.: The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, *Journal of Visual Languages and Computing*, Vol.7, No.1, pp.33–55 (1996).
8) Marks, J., et al.: Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation, *ACM SIGGRAPH '97*, pp.389–400 (1997).
9) Rekimoto, J.: The Information Cube: Using Transparency in 3D Information Visualization, *3rd Annual Workshop on Information Technologies & Systems*, pp.125–132 (1993).
10) Shneiderman, B., et al.: Ordered Treemap Layouts, *IEEE Information Visualization Symposium 2001*, pp.73–78 (2001).
11) 山口，伊藤，池端，梶永：階層型データ視覚化手法「データ宝石箱」とウェブサイトの視覚化，画像電子学会論文誌 Visual Computing 特集号，Vol.32, No.407, pp.407–417 (2003).
12) 山口，伊藤：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，情報処理学会グラフィクスと CAD 研究報告，2002-CG-108 (2002).
13) Yamaguchi, Y. and Itoh, T.: Visualization of Distributed Processes Using "Data Jewelry Box" Algorithm, *IEEE Computer & Graphics International 2003*, pp.162–169 (2003).
14) 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 グラフィクスと CAD 合同シンポジウム 2002，pp.1–6 (2002).

(平成 15 年 3 月 26 日受付)
(平成 15 年 9 月 5 日採録)

山口 裕美（正会員）

1977 年生．2001 年お茶の水女子大学大学院人間文化研究科数理・情報科学専攻博士前期課程修了．同年日本アイ・ビー・エム（株）入社．現在日本アイ・ビー・エム（株）東京基礎研究所副主任研究員．ビジュアリゼーション，ウェブサービス等に興味を持つ．

伊藤 貴之（正会員）

1968 年生．1992 年早稲田大学大学院理工学研究科電気工学専攻修士課程修了．同年日本アイ・ビー・エム（株）入社．現在日本アイ・ビー・エム（株）東京基礎研究所主任研究員．京都大学学術情報メディアセンター COE 研究員（客員助教授相当）兼任．博士（工学）．Visualization, CAD, CAE, CG, 分散処理システム，ネットワークセキュリティ等に興味を持つ．IEEE, ACM, 電子情報通信学会，芸術科学会各会員．