

平安京ビューII ～ 階層型時系列データのシームレスな視覚化手法

伊藤 貴之[○] 山口 裕美 小山田 耕二

(京都大学大学院情報学研究科, 日本アイ・ビー・エム(株)東京基礎研究所,

京都大学高等教育研究開発推進センター)

HeiankyoView II: Seamless Visualization of Time-Series Hierarchical Data

Takayuki ITOH, Yumi YAMAGUCHI, Koji KOYAMADA

itot@computer.org, yyumi@jp.ibm.com, koyamada@kudpc.kyoto-u.ac.jp

ABSTRACT Visualization is very useful for various large-scale computation areas. One of the authors has reported a hierarchical data visualization technique, Data Jewelry Box, and its extension which can smoothly represent time-series data. We also presented a new hierarchical data visualization technique, HeiankyoView, which is an algorithm improvement of Data Jewelry Box. This report presents the extension of HeiankyoView, which smoothly represents time-series data. As the extension of Data Jewelry Box did, this extension also consumes the ideal positions of nodes, and places the hierarchical data onto display spaces. Experimental results of this report show that this extension improves the smoothness of visualization results against Data Jewelry Box.

Keywords: Visualization, Hierarchical Data, Time-Series Data, Rectangle Packing.

1. はじめに

階層型データの視覚化手法は、情報視覚化の研究分野の中でも、特に活発に研究されている分野のひとつである。代表的なものとして、以下のような手法がある。

- 木構造を表現した手法。Hyperbolic Tree [Lam96] など。
- 画面空間の再帰分割による手法。Treemap [Joh91]など。
- 3次元空間で入れ子構造を構築し、半透明表示する手法。Information Cube [Rek93] など。

伊藤らは2次元空間で入れ子状に階層構造を構築する階層型データ視覚化手法「データ宝石箱」[Ito01]を報告し、続いてこれを階層型時系列データ向けに拡張した手法「データ宝石箱 II」[Yam02]を報告している。また、これらの画面配置結果の他手法[Bed02]に対する優位性を、実行結果をもって示している[Ito04]。

伊藤らは一方で、「データ宝石箱」の画面配置アルゴリズムを大きく改善する手法「平安京ビュー」[Ito03]を報告し、ネットワーク不正アクセス検出や、科学技術計算のパラメータ最適化 GUI などの目的で実用化を進めている。

本報告は、「平安京ビュー」を階層型時系列データ向けに拡張した手法「平安京ビューII」を報告する。本手法は「平安京ビュー」と同等の画面配置アルゴリズムを採用し、かつ「データ宝石箱 II」と同様に「テンプレート」という追加データ構造を用いている。テンプレートとは

階層型データを構成するノードの理想画面位置を記述したデータである。階層型時系列データをシームレスに視覚化するために、本手法では直前時刻における画面配置結果をテンプレートに記述し、それを参照しながら直後時刻における階層型データを画面配置する。本報告では、「平安京ビューII」による画面配置結果が、関連手法の画面配置結果よりもシームレスであることを、実験結果をもって示す。

2. 長方形の入れ子構造による階層型データの表現

「データ宝石箱」「平安京ビュー」とともに、階層型データを構成する葉ノードをアイコンで表現し、枝ノードを入れ子状の長方形の枠で表現することで、大規模な階層型データ全体を一画面に表示する。図1に、階層型データの視覚化例を示す。

このような視覚化を実現するために、「データ宝石箱」「平安京ビュー」とともに、階層型データを表現する長方形群に対して、以下の条件を満たすように画面配置している。

[条件 1] すでに配置されている長方形と干渉しない位置に長方形を配置する。

[条件 2] [条件 1]を満たす位置のうち、配置面積の拡大量が最小である位置に長方形を配置する。

以上の条件を満たす位置を高速に探し出すために、「データ宝石箱」「平安京ビュー」とともに、長方

形の候補位置を複数算出し、その中から最適な位置に長方形を配置する。「データ宝石箱」では、既に画面配置された長方形の中心点を連結する Delaunay 三角メッシュを生成し、この三角メッシュを参照しながら長方形の候補位置を算出する。「平安京ビュー」では、既に画面配置された長方形の稜線の延長線を用いて画面空間を格子状に分割し、この格子構造を参照しながら長方形の候補位置を算出する。各々の詳細については[Ito01][Ito03]に記述されている。著者らの実験では、「平安京ビュー」は「データ宝石箱」に対して、処理時間と画面占有面積を改善していることがわかっている。

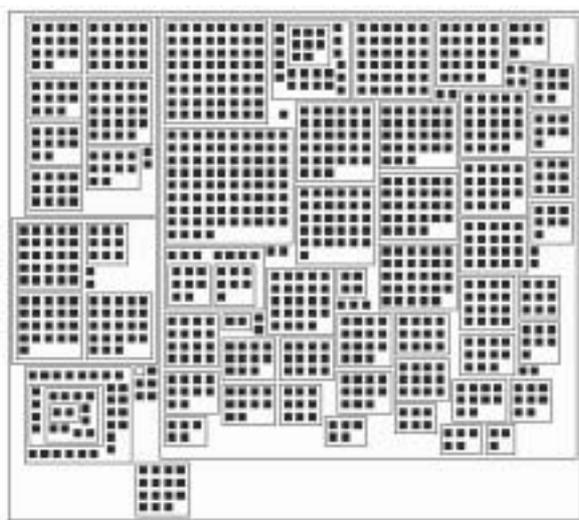


図 1 階層型データの視覚化例。

3. 「テンプレート」の導入による階層型時系列データのシームレスな視覚化

「データ宝石箱 II」では、階層型時系列データをシームレスに視覚化するために、テンプレートという追加データ構造を用いている。テンプレートとは階層型データを構成するノードの理想画面位置を記述したデータである。階層型時系列データをシームレスに視覚化するために、本手法では直前時刻における画面配置結果をテンプレートに記述し、それを参照しながら直後時刻における階層型データを画面配置する。

「データ宝石箱 II」では、2章で紹介した[条件 1][条件 2]に加えて、以下の条件を満たすように、階層型データを表現する長方形群を画面配置する。

[条件 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。

以上の条件を満たす長方形の位置を発見するために、「データ宝石箱 II」では、以下の手順で長方形の位置を 1 個ずつ決定する。

- (1) 画面空間内に、複数の候補位置を設定する。
- (2) 候補位置ごとに、以下の(2-1)(2-2)の処理を適用する。
 - (2-1) 長方形を配置した際に、既に配置されている他の長方形と重なるようであれば、この位置は[条件 1]を満たさないので、他の候補位置を選ぶ。
 - (2-2) 他の長方形と重ならないようであれば、評価値 $aD+bS$ を算出し、その値が過去最小であればその候補位置を仮の位置とする。
- (3) 全ての候補位置での処理が終了した時点で、「仮の位置」とされている位置に長方形を配置し、Delaunay 三角メッシュを更新する。

ここで a, b はユーザーが指定する正の定数、 D はテンプレートに記述された理想位置の距離、 S は長方形の配置による画面領域の面積拡大量とする。つまり $aD+bS$ 値が最小である位置は、[条件 2][条件 3]を両立する位置である、と判断することができる。

「データ宝石箱 II」では、まず面積最大である長方形 R_l を配置し、続いてテンプレートに記述された理想座標値間の距離が R_l に近い順に、1 個ずつ長方形を選択して配置する。この配置順は、近隣順に長方形を配置することで、画面上の不要な隙間の発生を避ける可能性が高い、という経験則に基づいて採用されている。

以上の詳細は[Yam02][Ito04]に記述されている。[Ito04]では、「データ宝石箱 II」が関連手法[Bed02]と比較してシームレスな階層型時系列データの視覚化を実現していることを、数値評価により実証している。

4. 平安京ビューII ~ 平安京ビューへのテンプレートの導入

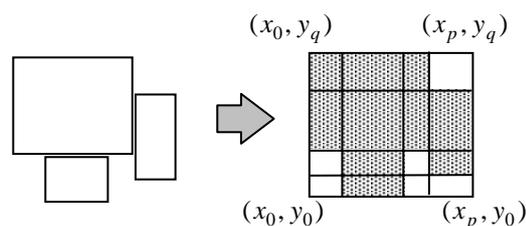


図 2. (左)すでに配置されている長方形群の例。(右)すでに配置されている長方形の辺を延長した線分で画面空間を格子状に分割した例。灰色に塗られていない格子領域の四隅を、次に配置する長方形の候補位置とする。

「平安京ビューII」は「平安京ビュー」と同様に、階層型データを表現する長方形群を 1 個ずつ画面配置する。長方形群の配置順は「データ宝石箱 II」と同様である。まず面積最大である長方形 R_l を配置し、続いてテンプレートに記述された理想座標値間の距離が R_l に近い順に、1 個ずつ長方形を選択して配置する。

また「平安京ビューII」は「平安京ビュー」と同様に、既に配置されている長方形の稜線の延長線を用いて画面空間を格子状に分割し、この格子構造を参照しながら長方形の候補位置を算出する。長方形の候補位置は、各格子領域の中で他の長方形に既に占有されていない格子領域の四隅であるとする（図2参照）。

ここで「平安京ビューII」は「データ宝石箱 II」と同様に、テンプレートに記述された理想座標値を参照しながら各々の長方形の位置を決定する。このとき各々の長方形に対して、以下の手順で位置を決定する。

(1) 画面空間を分割してできる格子領域の中から、以下の(1-1)(1-2)を満たすものだけを選択して、その四隅を候補位置とする。

(1-1) 他の長方形に占有されていない格子領域。

(1-2) これから配置する長方形の理想座標値に十分近く位置する格子領域。

(2) 候補位置ごとに(2-1)(2-2)の処理を適用する。

(2-1) 長方形を配置した際に、既に配置されている他の長方形と重なるようであれば、この位置は[条件1]を満たさないので、他の候補位置を選ぶ。

(2-2) 評価値 $aD+bs+cT$ を算出し、その値が過去最小であればその候補位置を仮の位置とする。

(3) 全ての候補位置での処理が終了した時点で、「仮の位置」とされている位置に長方形を配置し、格子分割データを更新する。

ここで a, b, D, S の意味は「データ宝石箱 II」と同様である。 c は a, b と同様に、ユーザーが指定する正の定数である。 T は候補位置に隣接する最大3個の格子領域(図3参照)のうち、既に他の長方形が占有している格子領域の個数 $m(m=0, 1, 2, 3)$ によって決まる正数であり、筆者らの実装では以下の値を用いている

$$T=0.0(m=3), 1.0(m=1), 2.0(m=2), 3.0(m=0)$$

これは「長方形に占有されている領域形状を単純化したほうが空領域を減らせる可能性が高い、という経験則から、占有領域の頂点を増やさない位置に高い優先度を置きたい」という考えに基づいて設定された値である。

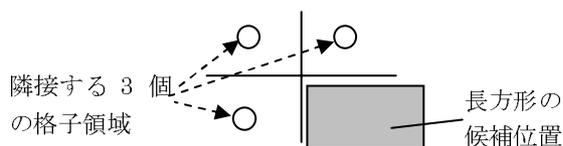


図3 長方形の候補位置に隣接する3個の格子領域。

5. 実行例

「平安京ビューII」を実行した例を図4に示す。この例では、時系列に沿ってノード数が微量だけ増減している階層型データを題材にして、以下のような手順で階層型データを視覚化した。

- (1) 「平安京ビュー」により1個目のデータを視覚化した。
- (2) (1)の画面配置結果をテンプレートに記述して、これを参照しながら「平安京ビューII」により2個目のデータを視覚化した。
- (3) (2)の画面配置結果をテンプレートに記述して、これを参照しながら「平安京ビューII」により3個目のデータを視覚化した。

図3を見てもわかるとおり、個々のデータを表現するデータ要素の画面上の移動量は小さく、階層型時系列データのシームレスな表現を実現できていることがわかる。

続いて「平安京ビューII」と関連手法の画面配置結果を数値評価した結果を示す。この数値評価では、階層型データの各階層を表現する長方形領域の4頂点の座標値が $(-1, -1)$ $(1, -1)$ $(-1, 1)$ $(1, 1)$ になるように座標値を正規化し、

(a) 1個目と2個目の視覚化結果

(b) 2個目と3個目の視覚化結果

において対応する各々のノード間の距離 D を算出した。 D 値の平均値および最悪値が小さいほど、画面配置結果の直前時刻との差分が小さく、シームレスな画面配置を実現している、ということがいえる。

表1および表2に、 D 値の平均値および最悪値を、上記(a)(b)に対して算出した結果を示す。関連手法として「データ宝石箱 II」の他に、階層型データの葉ノードをアイコン表示するという点で本手法と類似している Quantum Treemap [Bed02] の2種類の実装を用いた。

表1 実行結果から算出した D 値の平均値

手法	平均値(a)	平均値(b)
平安京ビューII	0.0657	0.0734
データ宝石箱 II	0.1209	0.1176
Quantum Squarified Treemap	0.4824	0.4827
Quantum Strip Treemap	0.4759	0.4729

表2 実行結果から算出した D 値の最悪値

手法	最悪値(a)	最悪値(b)
平安京ビューII	0.7073	0.6411
データ宝石箱 II	0.8137	0.8721
Quantum Squarified Treemap	1.3638	1.3831
Quantum Strip Treemap	1.3655	1.3255

以上の結果から、「平安京ビューII」が関連手法に比べて、画面配置結果の直前時刻との差分が小さく、シームレスな画面配置を実現できていることを示した。

6. まとめ

本報告では、階層型データの新しい視覚化手法「平安京ビュー」に、「データ宝石箱 II」で導入されているテンプレートという概念を導入することで、階層型時系列データをシームレスに視覚化する手法を提案した。また関連手法に対する優位性を実験結果により示した。

「平安京ビューII」では、時系列データを反復的に視覚化する過程で、徐々に画面占有面積が拡大する傾向がある。これを改善する対策として、

- 4章で示した定数 a, b, c を動的に調節する
 - 画面占有密度が一定以下になったら、いったんテンプレートの参照を止めて画面配置を初期化する
- などが考えられる。これらの案の有効性を検証することを、今後の課題としたい。

また「平安京ビューII」は、テンプレートへの理想座標値を記述さえできれば、階層型時系列データ以外にも応用が可能である。筆者らは現在、この特性を活かした拡張手法「平安京ビューIII」を考案中である。次報ではこの「平安京ビューIII」を報告したい。

謝辞

日頃貴重な議論を頂く京都大学関係諸氏に感謝します。

参考文献

[Bed02] Bederson B., Schneiderman B., Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, *ACM Transactions on Graphics*, 21, 4, pp. 833-854, 2002.

[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&CAD 研究会, 2001-CG-104, 2001.

[Ito03] 伊藤, 小山田, 平安京ビュー ~ 階層型データを基盤状に配置する視覚化手法, 可視化情報学会第9回ビジュアルイノベーションカンファレンス, 2003

[Ito04] Itoh T., Yamaguchi Y., Ikehata Y., Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle Packing Algorithm, *IEEE Transactions on Visualization and Computer Graphics*, 10, 3, pp. 302-313, 2004.

[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.

[Lam96] Lamping J., Rao R., The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, *Journal of Visual Languages and Computing*, 7, 1, pp. 33-55, 1996.

[Rek93] Rekimoto J., The Information Cube: Using Transparency in 3D Information Visualization, *Third Annual Workshop on Information Technologies & Systems*, pp. 125-132, 1993.

[Yam02] 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ックス&CAD 研究会, 2002-CG-108, 2002.

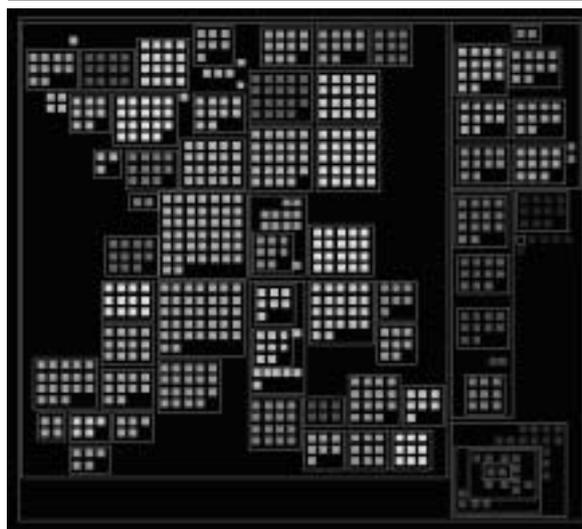
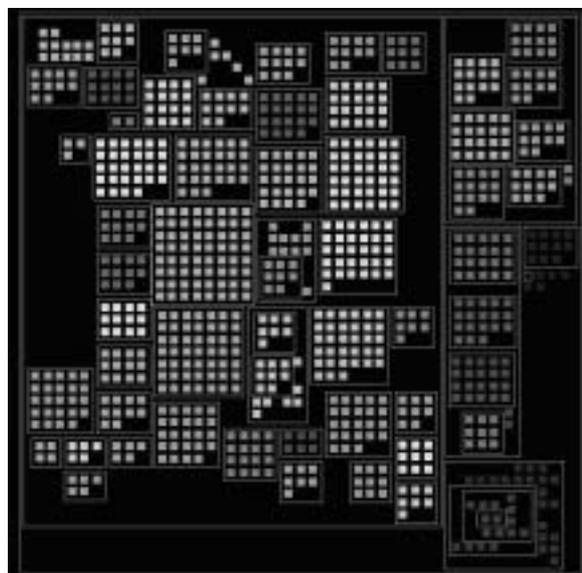
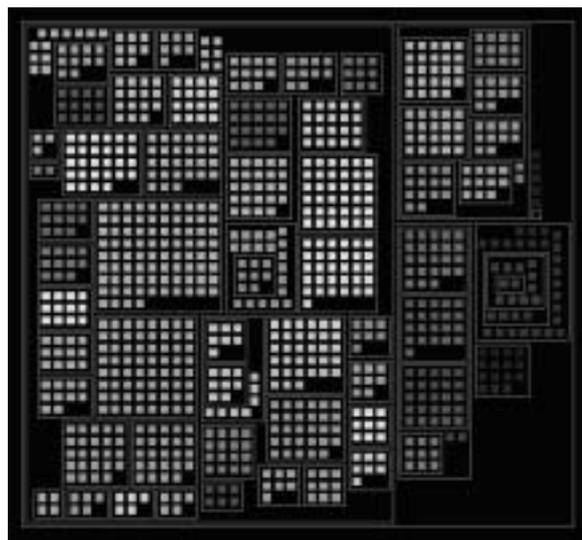


図4(上)1個目の階層型データの視覚化結果。(中)2個目の階層型データの視覚化結果。(下)3個目の階層型データの視覚化結果。