3C－04

折り返し鎖形式の空間分割手法を用いた統計データの視覚化手法の提案*

池端 裕子　伊藤 貴之　梶永 泰正　山口裕美†
日本アイ・ビー・エム(株) 東京基礎研究所‡

1.はじめに

本報告は、円グラフや帯グラフのように、画面空間を“階級の並び（ランキング）“にしたがって長方形領域の集合に分割表現する、「空間分割型」の統計データの視覚化技術の改善手法を提案する。

本報告が対象としている大規模な統計データは、身の周りに非常に多く存在する。筆者らは、例えばウェブサイト群のアクセス数のデータや更新日時を集計したデータ、計算機のファイルシステムの利用頻度を表すデータ、会社の各事業所のサイトごとに集計した利益率を表したデータ、並列計算機のプロセス管理の使用率を表すデータ等、なんらかの属性値を持つデータを対象としている。本報告で提案する視覚化技術は、大規模統計データに対する傾向分析、理解、整理、などの目的で有用であると考えられる。

統計データを階級化したときに、階級数が非常に大きくなる事例、あるいは階級値が非常に小さくなる事例は多々ある。下の表1は、1991年から2001年までに更新されたウェブページを更新月ごとに区切って、階級化したデータの例である。この例では、階級が120個以上あり、非常に多い。また階級番号118の階級値は3であり、他の階級値に比べて非常に小さい。

なお本報告では、階級値と区間値の組を「階級」と呼ぶこととする。

表1　統計データを更新月で階級化したときの区間値と階級値の関係

階級 番号	…	118	119	120	…
更新日時の区間値	…	2001/01/01 | 2001/01/31	2001/02/01 | 2001/02/28	2001/03/01 | 2001/03/31	…
階級値（ウェブページ数）	…	3	125	75	…

空間分割によって統計データを表現する最も身近な手段は、円グラフや帯グラフである。円グラフや帯グラフは階級を一列に並べる。そのため、非常に多くの区間を持つデータの場合、帯グラフでは非常に細長い空間を、円グラフでは非常に広い円形空間

---

*Folded-Chain Diagrams: A Statistical Data Visualization Technique Using a Folded-Chain Style of Space-Filling Scheme
†Yuko Ikehata, Takayuki Itoh, Yasumasa Kajinaga, and Yumi Yamaguchi
‡IBM Research, Tokyo Research Laboratory

を必要とする。逆に、多くの統計データを同時に一画面に表示するときは、１個の統計データに与えられる空間が狭くなる。このように、階級値の小さいデータを表現する形状領域が画面上で細長く分割されるので、見落とされたり、ディスプレイの解像度によってはつぶれて表示されなかったり、という問題が生じる。このような条件下では、帯グラフ、円グラフともに、少ない階級数しか表現できない。

一方、階層を構成する統計データを、2 次元的な空間分割によって表現する視覚化手法として、Treemap 法[1][2][3]が提案されている。Treemap 法は、まず長方形領域を縦に分割し、続いて各々の領域を横に分割し、必要があればさらに各々の領域を縦に分割しという反復処理によって、帯グラフが入れ子になったような領域分割を実現する。

Treemap 法も、階級値の小さい階級を表現する形状領域が画面上で細長く分割されるので、見逃されたり、ディスプレイの解像度によってはつぶれて表示されなかったり、という問題が生じる。

文献[2]の著者らは近年、階層構造でない統計データに対しても、Treemap 法のような２次元的な領域分割を用いて、階級値に比例した面積の長方形群で長方形領域を分割する表現手法を提案した[4][5][6]。

図１(左)に示すような SquarifiedTreemap 法［4］や ClusteredTreemap 法[5]は、階級値の大きい順に階級を配置する。これらの手法は、できるだけ正方形に近い形状の長方形群で長方形領域を分割することを目的としている。この手法は、階級を表す長方形を正方形に近い形状で表現するので、

[長所 1]階級値の 小さい階級まで見逃したくないときに、階級値の小さい階級が細長くつぶれて見えなくなることを防ぐ。

という点においてメリットがある。しかしその反面、

[短所 1] 長方形の位置関係は、階級の区間値とはまったく無関係となるので、近隣する区間値をもつ階級間の隣接関係の把握が困難である。また、

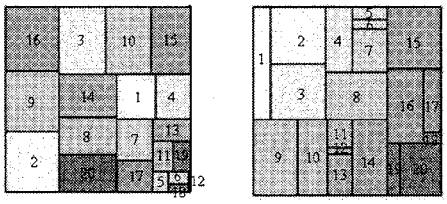

図１（左）Squarified Treemap。（右）Ordered Treemap。各階級に相当する長方形領域は、階級値に比例した面積をもつ。番号はその階級の区間値の順番をあらわす。両者とも、隣接する区間値をもつ階級が画面上で隣接することを保証していない。

階級の並び（ランキング）を把握することが容易ではない。

という問題点がある。

図１(右)に示すような OrderedTreemap 法［6]は、階級を区間値順に、しかも長方形が正方形に近い形状となるように配置する。この手法は、

[長所 2] 長方形の位置関係は、階級の区間値の順となるので、近隣する区間値をもつ階級間の視覚的な比較がしやすい。

というメリットがある。しかしその反面、

[短所 2] 長方形の形状は、SquarifiedTreemap や ClusteredTreemap ほど正方形に近い形状とはならない。

[短所 3] 長方形の位置関係は、階級の区間値の順となるものの、隣接する区間値をもつ階級どうしが離れて配置されることも随所に見られる。そのため、階級の並び（ランキング）を把握することが容易ではない。

という問題点がある。

本報告では、以上の従来手法の短所を解決し、以下の条件を満たすような、2 次元的な空間分割による統計データの視覚化手法を提案する。

[条件 1] 階級値の小さな階級を表現する長方形形状が細長くなり、見逃されたり、画面上でつぶれて見えなくなったりすることを避ける。

[条件２] 区間値順の位置関係で長方形を配置する。

[条件３] 隣接する区間値をもつ階級が必ず画面

上で隣接するような位置関係で、長方形を配置する。

第２章では提案する視覚化手法の特徴、第３章では、配置アルゴリズムの提案、第４章では上記のアルゴリズムを用いたデータ配置結果を示し、従来手法と比較する。

2. 本手法の特徴

本手法では、各区間値に分類された階級を、その階級値の大きさに比例する面積の長方形領域で表現する。与えられた画面空間を、これらの長方形領域で２次元的に分割することで、統計データを表現する。本手法は以下の２つの特徴により、統計データの階級の並びを画面上で一望できるような視覚化を実現している。

2.1「折り返し鎖形式」の配置手法

図２に、折り返し鎖形式の配置手法の概念図を示す。本報告で提案する手法は、まず画面空間を上から下、下から上..というように折り返しながら、隣接する区間値をもつ階級が必ず隣接するように長方形領域を配置する。その長方形領域の空間配置は、折り返しながら連結された長い鎖のようなスタイルなので、本報告ではこれを「折り返し鎖形式」と呼ぶ。この形式により本手法では、**[条件２]**と**[条件３]**を満たすような配置結果を実現する。

2.2 ユーザーにとって重要な意味をもつ階級を優先する配置手法

従来手法のうち SquarifiedTreemap 法，ClusteredTreemap 法は、階級を階級値の大きさ順に配置し、かつ配置結果の位置関係も階級値の大きさ順となる。また、OrderedTreemap 法は、階級を区間値順に配置し、かつ配置結果の位置関係も階級の区間値順となる。

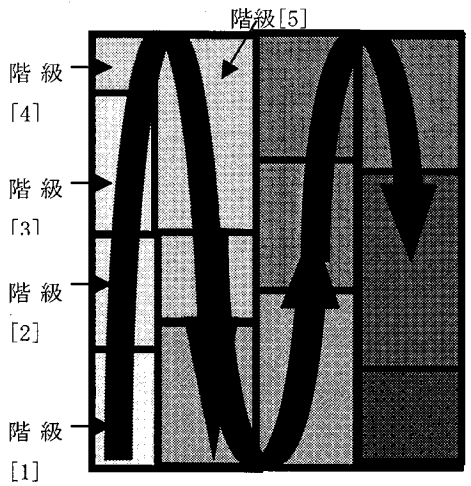

図２「折り返し鎖形式」の配置手法

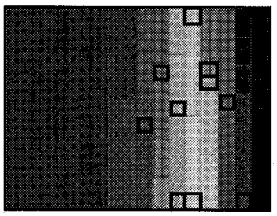

図３階級値の小さな階級を表す長方形が正方形に近い形状になるように、長方形を配置した結果。面積の小さい長方形が画面上でつぶれて見えなくなるのを防いでいる。

それに対して本報告では、図３で示すように、ユーザーが階級値の小さい階級を重要視したいときは階級値の小さい階級から順に、階級値が大きい階級を重要視したいときは階級値の大きい階級から順に配置し、先に配置した長方形の形状を優先的に正方形に近づけるアルゴリズムを提案する。このとき、階級値の小さな階級から配置することで、階級値の小さな階級を優先的に正方形に近づけ、細長くなることを防ぐことができるので、**[条件１]**を満たすことができる。

なお、本手法を適用すると、配置結果の位置関係は階級の区間値順であり、かつ折り返し鎖形式を実現することができる。

3. 配置アルゴリズムの提案

3.1 前提条件

n 個の階級で構成される統計データを考える。この階級の階級値を区間値順に $r_1,...,r_n$ とし、その合計値を r_{all} とする。また、与えられた長方形領域を A_{all} とする。このとき、各階級を表現する長方形の面積 $A_1,...,A_n$ は、式(1)により求めることができる。

$$A_k = \frac{r_k}{r_{all}} A_{all} \quad \text{ただし} \quad r_{all} = \sum_{i=1}^{n} r_i \qquad \cdots(1)$$

また、$A_1,...,A_n$ を小さい順（または大きい順）に並べ替えたものを $A_{s_1},...,A_{s_n}$ と表す。本報告では、このように面積で並べ替えた順に長方形を配置するものとする。

3.2 階級の初期配置

以下、階級値の大きい順に階級を配置する場合を例にして、本手法のアルゴリズムを説明する。本手法では、まず階級を構成する長方形を、「長方形 1 個が帯領域 1 個を構成する」と想定して、図 4 のように配置する。幅および高さの算出方法は以下の通りである。長方形 A_k の幅および高さを (w_k,h_k) とし、与えられた長方形領域 A_{all} の幅および高さを (w_{all},h_{all}) とすると、(w_k,h_k) は式(2)により算出される。

$$w_k = \frac{A_k}{A_{all}} w_{all} \qquad h_k = h_{all} \qquad \cdots(2)$$

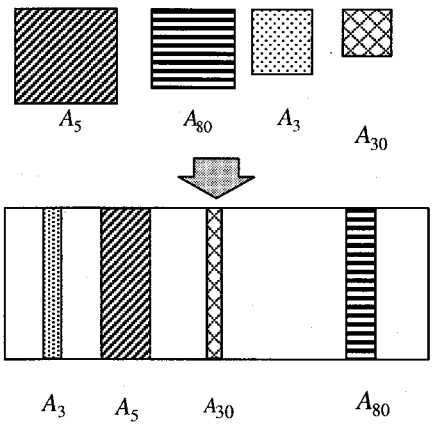

図 4 階級値が大きい順（または小さい順）の配置。階級の位置関係は、区間値順に相対的におく。

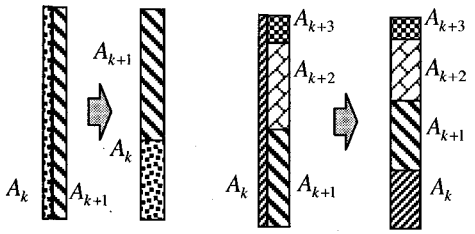

図 5 いま配置した階級 A_k の近隣帯領域への併合

3.3 階級の隣接帯領域への併合

続いて本手法では、近接した帯領域を併合することで、配置した階級の長方形領域の形状をより正方形に近づけることを考える。まず、いま配置した階級に隣接する区間値をもつ階級が、すでに配置されているかチェックする。図 5 で示すように、長方形 A_k を配置したときに、A_{k-1} あるいは A_{k+1} がすでに配置されているかを確認する。すでにいずれかが配置されているときは、A_{k-1} あるいは A_{k+1} を含む帯領域に A_k を併合すべきかどうか判定する。ここで、併合の判定の指標として本手法では、長方形の幅と高さの比（アスペクト比）を用いる。長方形 A_k のアスペクト比を a_k とすると、a_k は式(3)により算出される。

$$a_k = \begin{Bmatrix} w_k / h_k (w_k > h_k) \\ h_k / w_k (w_k < h_k) \end{Bmatrix} \qquad \cdots(3)$$

仮に、$A_{k+1},...,A_l (k<l)$ で構成される帯領域に A_k を併合するかどうか判定するとする。帯領域の現在の幅を w_{before} とすると、A_k を併合したとき後の帯領域の幅 w_{after} は式(4)により算出される。

$$w_{after} = \frac{A_{after}}{A_{before}} w_{before}$$

$$\text{ただし} \quad A_{before} = \sum_{i=k+1}^{l} A_i, A_{after} = \sum_{i=k}^{l} A_i \qquad \cdots(4)$$

また、帯領域中にすでに配置されている長方形 $A_{k+1},...,A_l (k<l)$ のうち、最も面積の大きな長方形を A_{imp} とする。（この例では階級値の大きい順に配置しているので最も面積の大きい長方形を選んだが、階級値の小さい順に配置しているときは最も面積の小さい長方形を選ぶ。）このとき、A_k を併合する前

の A_{imp} の高さを h_{before} とし、併合した後の A_{imp} の高さを h_{after} とすると、

$$h_{before}=\frac{A_{imp}}{w_{before}},\quad h_{after}=\frac{A_{imp}}{w_{after}} \quad \cdots(5)$$

である。以上の数式によって算出される w_{before}, w_{after}, h_{before}, h_{after} を用いて、併合前の A_{imp} のアスペクト比 a_{before} および併合後の A_{imp} のアスペクト比 a_{after} を算出する。$a_{before}>a_{after}$ ならばアスペクト比が向上するので併合する。逆に $a_{before}\le a_{after}$ ならアスペクト比は向上しないので併合しない。

本手法では、帯領域の併合処理は頻繁に発生するが、そのたびに先に配置された重要度の高い階級の形状が評価され、その形状が正方形に近づくように併合判定が行われるので、重要度の高い階級の形状が正方形に近づく。

3.4 2帯領域の再編成

3.3章では、いま配置した階級 A_k と、すでに配置されている隣接帯領域 $A_{k+1},...,A_l(k<l)$ の併合判定に関して記述した。この判定と同時に本手法では、$A_k,...,A_l(k<l)$ を2帯領域に再編成することで、長方形領域の形状をより正方形に近づけることを考える。

図6に示すように、仮に、$A_{k+1},...,A_l(k<l)$ で構成される帯領域の隣に A_k を配置したときに、これらの階級を $A_k,...,A_m(k<m)$ および $A_{m+1},...,A_l(m<l)$ の2帯領域に再編成するかどうか判定するとする。帯領域の現在の幅を w_{before} とすると、$A_k,...,A_m(k<m)$ で構成される帯領域の幅 w_{after1}、および $A_{m+1},...,A_l(m<l)$ で構成される帯領域の幅 w_{after2} は、式(6)により算出される。

$$w_{after1}=\frac{A_{after1}}{A_{before}}w_{before},\quad w_{after2}=\frac{A_{after2}}{A_{before}}w_{before}$$

ただし

$$A_{before}=\sum_{i=k+1}^{l}A_i, A_{after1}=\sum_{i=k}^{m}A_i, A_{after2}=\sum_{i=m+1}^{l}A_i$$

... (6)

また、帯領域中にすでに配置されている長方形 $A_{k+1},...,A_l(k<l)$ のうち、最も面積の大きなものを A_{imp} とする。（この例では階級値の大きい順に配置しているので最も面積の大きい長方形を選んだが、

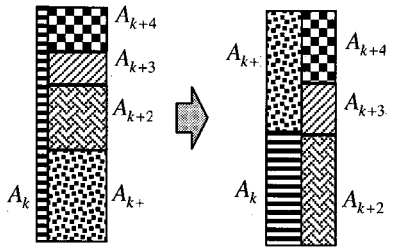

図6 いま配置した階級 A_k を含む2帯領域の再編成

階級値の小さい順に配置しているときは最も面積の小さい長方形を選ぶ。）このとき、A_k を配置する前の A_{imp} の高さを h_{before} とし、配置した後の A_{imp} の高さを h_{after} とすると、

$$h_{before}=A_{imp}/w_{before}, h_{after}=A_{imp}/w_{after}$$

ただし $w_{after}=w_{after1}(k<imp\le m),\ w_{after}=w_{after2}(m<imp<l)$ ... (7)

である。以上の数式によって算出される w_{before}, w_{after}, h_{before}, h_{after} を用いて、併合前の A_{imp} のアスペクト比 a_{before} および併合後の A_{imp} のアスペクト比 a_{after} を算出する。$a_{before}>a_{after}$ ならばアスペクト比が向上するので併合する。逆に $a_{before}\le a_{after}$ ならアスペクト比は向上しないので併合しない。

以上の処理を、$k+1<m<l$ の範囲で m を変動させながら反復し、最も良好なアスペクト比が算出される状態で帯領域の再編成を実現する。

本手法では、帯領域の再編成処理は頻繁に発生するが、そのたびに先に配置された重要度の高い階級の形状が評価され、その形状が正方形に近づくように再編成判定が行われるので、重要度の高い階級の形状が正方形に近づく。

3.5「折り返し鎖形式」の配置パターンの実現

「折り返し鎖形式」の配置パターンを実現するためには、例えば左から奇数番目の帯領域では階級を下から上へ、偶数番目の帯領域では逆に階級を上から下へ、というように配置する必要がある。しかし本手法では、各々の帯領域が左から何番目の帯領域になるか、すべての階級の配置を終えるまで確定で

きない。そこで本手法では、すべての階級の配置を終えてから、図7で示すように奇数番目の帯領域では下から上へ、偶数番目の帯領域では上から下へ、という配置順になるように階級の座標値を再算出する。

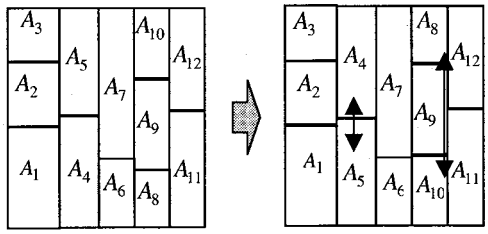

図7「折り返し鎖形式」の配置パターンの実現

4. 実行例

本手法を用いて得られる長方形のアスペクト比を評価した。

図8, 9は、あるウェブサイトを構成するウェブページ群、及びそこから直接リンクされているウェブページ群を視覚化したものである。実験で用いた統計データは33個のグループで構成され、階級は総計127個であった。本手法は階級値の小さいデータから順に配置して結果を得た。

本手法と従来手法（OrderedTreemap-likeな実装）の2種類のアルゴリズムを用いて比較したところ、従来手法では図9の矢印部分の階級は細長く、見逃しやすいのに対して、本手法では図8矢印部分の階級の形状はより良好であることがわかる。

階級値の小さな階級群（階級値のシェア率が 2% より小さいデータを「階級値の小さな階級」とする）の中で最悪なアスペクト比と、アスペクト比の平均値について2種類のアルゴリズムを比較した結果、以下のことがわかった。

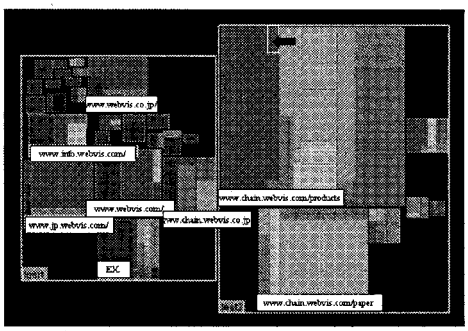

図8 本手法を用いて、階級値の小さい階級から配置した例。同じ階級を表す図9の矢印部分と比較すると、形状が正方形に近く良好である。

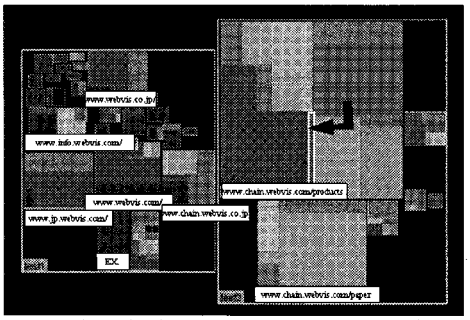

図9 従来手法（OrderedTreemap-like な実装）による画像例。同じ階級を表す図8の矢印部分と比較すると、形状が細長く、画面上でつぶれて見えなくなりやすい。

- 階級値の小さな階級群の中で、本手法を用いたときの最悪なアスペクト比の値は 3.87 であり、OrderedTreemap を用いた場合 22.6 と比較して、かなり良好な値であった。
- 階級値の小さな階級群の中で、本手法を用いたときのアスペクト比の平均値は 1.73 であり、OrderedTreemap を用いた場合 2.39 と比較して、良好な平均値であった。

以上の例からわかるように、本手法は従来手法に比べて、階級値の小さな階級が画面上でつぶされたり見逃されたりしないように、面積の小さい長方形領域の形状を優先的に正方形に近づけることに成功していることがわかる。

5. むすび

本報告では、与えられた長方形領域を縦長の帯領域に分割し、その帯領域を下から上に、続いてその隣の帯領域を上から下に … というような「折り返し鎖形式」で配置した長方形の集合により、統計データを表現する視覚化手法を提案した。更に、階級値の小さな(または大きな)階級から順に配置して、先に配置した長方形の形状を優先的に正方形に近づけるアルゴリズムを提案した。

本手法の特徴は、以下の通りである。

- 隣接する区間値をもつ階級が必ず画面上で隣接するような位置関係で、階級を配置できるので、区間値の隣接する階級を視覚的に比較しやすい。
- 階級値の小さい階級から順に配置し、それを表現する長方形を正方形に近い形状で配置することにより、階級値が小さい階級が画面上で細長くつぶれて見えなくなるという従来手法の問題点を改善できる。

また今後の課題として、以下のようなことを検討中である。

- さらに大きなデータに対する視覚化の実験と検討。
- 時間変化を伴う統計データのシームレスな視覚化手法の開発。
- 本手法を用いたGUIやシステムの構築。

謝辞

多くの助言と協力を下さった、日本アイ・ビー・エム(株)東京基礎研究所梶谷浩一氏、青野雅樹氏、井上恵介氏、山田敦氏、土井淳氏に、感謝の意を表する。

参考文献

1. Johnson B. and Shneiderman B., Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures, *In Proceeding of the 2nd International IEEE Visualization Conference*, pp. 284-291, 1991.
2. Shneiderman, B., Tree Visualization with Tree-Maps: 2-d Space-Filling Approach. *ACM Transactions on Graphics*, 11(1), 1992, pp. 92-99.
3. http://www.cs.umd.edu/hcil/treemaps/
4. Bruls, D.M., C. Huizing, J.J. van Wijk. Squarified Treemaps. In: W. de Leeuw, R. van Liere (eds.), *Data visualization 2000, Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization*, 2000, pp. 33-42.
5. Cluster Treemap M. Wattenberg. *Map of the Market.* http://smartmoney.com/marketmap/, Smart-Money.com, 1998.
6. OrderedTreemap ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.htm.