

大規模階層構造データの識別情報を用いた可視化手法

桑原 明栄子† 牧野 光則†

Email:kuwahara@makino.ise.chuo-u.ac.jp, makino@m.ieice.org

近年の情報技術の発展により、何らかの識別情報を有する大規模な階層構造デジタルデータが重要な位置を占めている。この種のデータを容易に把握するために、さまざまな可視化手法・GUIが提案されている。代表的なGUIでは、まずデータの上位構造を表示し、続いてユーザの選択操作に基づき徐々に下位構造を表示する。一方、データを一目で把握するために一画面で全データを展開・表示する研究が進められている。このとき、各データの識別情報を併せて表示することは、データの的確な理解のために重要である。そこで、本稿では階層構造データが持つ識別情報に着目して、全データを一面表示する手法を提案する。提案手法では、対象データの構造と識別情報を基準にデータの順位を決定する。この順位により表示方法(位置、大きさ、色、形状など)を変化させる。これにより、識別情報も含むデータを可視化でき、情報の把握・確認をより容易にする。

A Visualization of Large Scale Hierarchical Data with Subsidiary Information

Meeko KUWAHARA† Mitsunori MAKINO†

Recently large scale hierarchical data with subsidiary information has been becoming important together with development of information technology. Various visualization techniques and GUIs have been proposed to let us understand easily this type of data. In typical case, a GUI displays top structure of the data firstly, followed by displaying lower structure. In another case, some visualization techniques, displaying all data in one screen, have been studied, which show us whole structure at a sight. It is meaningful for deeper understanding that not only the structure but also the subsidiary information can be visualized. In this paper, a visualization method of large scale hierarchical data is proposed, which can show all in one screen with their subsidiary information. In the proposed method, the structure and the subsidiary information are used for determination of order, position, shape, height and color. An evaluation of the proposed method is shown in a case of a type of webpage.

1 はじめに

近年の情報技術の発展により、階層構造によって管理されている大規模なデジタルデータが重要な位置を占めている。この種のデータを容易に把握するために、さまざまな可視化手法・GUI(Graphical User Interface)が提案されている。Windows エクスプローラ等の代表的なGUIでは、まずデータの上位構造を、続いてユーザの選択操作に基づいて徐々に下位構造を表示する手法が利用されている。この場合、表示される情報が限定される。一方、データ

全体を一目で把握するために、全データを一画面に展開・表示する研究が進められている[1]。このとき、データの的確な理解のために、各データが保有する階層構造以外の識別情報も併せた表示が重要である。

伊藤らは、大規模階層型データを下位階層にいたるまで一面で表示する視覚化手法を提案した[2]。一階層を構成するデータ群を隙間なく配置して、それを長方形で囲むという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、階層型データ全体を画面空間に配置する。長方形群を干渉なく、かつ隙間なく配置できる位置を高速に検出するために、長方形の中心点群を連結するDelaunay 三角形メッシュを用いるアルゴリズムを提案した。伊藤らの手法は高速に大規模階層型データを下位階層にいたるまで一面で表示することを可能としている。しかし、識

† 中央大学大学院 理工学研究科 情報工学専攻

† Graduate School of Science and Engineering, Chuo University

別情報を順位によって表示を変化させてはいない。

そこで、本稿では階層構造データがもつ識別情報に着目して、全データを一画面表示する手法を提案する。また、使用データとして、Web データを使用し、それらの識別情報（ファイル拡張子・アクセスログ等）を利用する。

提案手法では、対象データの構造と識別情報を基準にデータの順位を決定する。この順位により表示（位置、大きさ、色、形状等）を変化させる。対象データの利用方法によって、識別情報も含むデータを可視化でき、情報の把握・確認をより容易にする。また、本稿においては、使用データの利用方法から、Web データを識別情報つきで可視化する場合の主たる目的が、時々刻々と変化するアクセス頻度や不正アクセスの有無の解析となり、対話型システムにする必要がある。そのため、最適配置よりも、処理時間に重点をおく。

2 提案手法

大規模階層型データを理解しやすく、下位階層にいたるまで一面で表示するために、本稿では、各データがもつ識別情報を利用し、その識別情報によって個々のデータの表示形態を変化させる。本稿では代表的な階層構造型データ例として、Web データを使用する。Web データの階層構造を管理することを考慮し、最適配置よりも処理時間の短縮を目的とした。そのため、ファイルを表現するアイコンのサイズ・配列を設定・制限した。

2.1 配置

各ファイルを表示する形式は円グラフの様な形態を使用する。

以下の手順で、各ディレクトリ、ファイルの配置を決定する。階層構造データの最上位階層にあるディレクトリは扇型で表される。(ファイルを表すアイコンの横の長さ Iw) \leq (ファイルを表すアイコンの縦の長さ Ih) と設定する。

1. 階層構造データの必要な情報を取得する。

2. (Ih の 2 倍の数) \times (中心から数えた円周の列数) を円の半径とする。
3. 円周上に Iw の 2 倍の長さ毎に配置していく。
4. 3. にしたがって、全てのファイル分のアイコンが入りきるまで 2. より円の円周を作成する。そのことにより、暫定的にアイコンを配置する円の円周の数を決定する。
5. 各ディレクトリの中にディレクトリに属さないファイルが存在するため、属さないファイルが入る区画を確保するために、属さないファイル群が入る区画を必要に応じて 1 つのディレクトリとして計算する。
6. 階層構造データの最上位階層にあるディレクトリの各ファイルを全てのファイル数から、比率を計算し、各最上位階層にあるディレクトリが占める扇型の区画の角度を決定する。
7. 階層構造データの最上位階層にあるディレクトリで最もファイル数が少ないディレクトリを 4. で暫定的に求めた円の中に扇型の角度が最小で表示できるように配置する。
8. 残りのディレクトリと各ファイルを 6. で求めた比率に基づき配置する。ファイルが比率によって求めた区画に入りきらない場合は列数を増やす。(7. で配置した最もファイル数が少ないファイルの配置により、比率がずれた場合はずれた比率を残りのディレクトリの比率に等分して反映させる。)
9. ディレクトリ内は、ディレクトリ、ファイルの種類、ファイル名の順にソートされる。
10. ファイルの種類は、設定ファイルを設け、その設定ファイルに記述した順によってソートされる。
11. 各ファイルは、区切られた区画の左外側から順に並び、次の列は右から、その次の列は左側からといったように、一列ごとに並び順を逆にする。

12. ディレクトリを表すアイコンを円の外側から配置し、表示する。
13. 1. で取得した情報の中から順位によって表示したい情報を取得し、その情報に基づき、最上位階層にあるディレクトリをソートする。

2.2 表示形式

Web データの各ディレクトリ、各ファイルがもつ識別情報を以下のように表示する。

- 順位： 識別情報から出した順位の高い最上位階層にあるディレクトリの円の外側に順位の方向に矢印を表示する。
- ディレクトリ： ディレクトリ毎に線で囲む。
- ディレクトリの種類： パーミッションの設定により、ディレクトリ毎に囲んでいる線の色を変化させる。
- ディレクトリの情報： ディレクトリの情報はディレクトリを表示している区画のファイルを表すアイコン以外の部分をクリックすると表示する。
- ファイル： ファイル毎にアイコンで表示する。
- ファイルの情報： ファイルの情報は各ディレクトリを表すアイコンをクリックすることにより表示する。
- ファイル拡張子： ファイル拡張子の種類によって、色を変化させる。
- ファイルアクセス数： アクセス数に基づいてファイルを表すアイコンの高さを変化させる。
- ファイルアクセス元： 特定のアクセス元に基づいてファイルを表すアイコンの形状を変化させる。
- ファイル不正アクセス： アクセス制限を行なっているファイルにおいて、不正な行動があったファイルのアイコンの上に球を表示させる。

2.3 提案手法の特徴

提案手法の主な特徴は以下の 4 点である。

- 識別情報に基づきアイコンの色や形を変化させ、また、ユーザが指定する識別情報に基づき表示順位を変更できる。このため、識別情報が多い Web データに対しても、十分な表現力をもつ。
- 最上位階層を円形表示しているため、視点変更しても違和感がない。
- 階層構造の可視化は、表示順序を変化させることを考慮し、全体の表示を円形、最上位階層を扇型とし、子階層を階層ごとに線で区切る形式とした。この表示形式により、最上位階層の順位の変更 (ソートする識別情報の変更、順位そのものの変化) の際、配置の再構築ならびに、表示形式の変更が不要となる。
- アイコンの配置には規定値を設定し、処理時間が短縮している。このため、アイコンは最適配置とはならないが、対話型可視化に適している。

一方、円の中心付近にはアイコンを原理上配置できない。また、長方形画面に対して円形表示することから、画面内にアイコンが表示されない領域が必ず存在する。そのため、対象データ数が増加した場合にアイコンが小さくなりやすい。

3 シミュレーション

本提案手法によって Web データを可視化した結果を図 1~6 に、動作環境を表 1 に、識別情報の表示形式を表 2 に示す。なお、可視化にあたっては OpenGL を利用し、ウィンドウの表示サイズならびに視点位置をマウスで変更可能とした。

今回使用した Web データは、実際のデータを参考にして作成したモデルである。また、紙面に掲載する場合の理解しやすさを考慮し、通常よりもファイル数・ディレクトリ数を少なく設定した (表 3)。また、必要な情報は既に取得済みであると仮定し、Web サーバからの情報取得はしていない。

表 1: 動作環境

CPU	インテル Pentium 4 2.40 GHz
OS	Windows XP
メモリ	512MB
ビデオボード	nVIDIA Geforce4 Ti4200

表示順位として今回指定したのは、最上位階層にあるディレクトリの名前順である。実験では、最上位階層に含まれるファイル群は名前なしのディレクトリに含まれるとみなしたので、順位が最高となっている。

図 1 は真上から見た図、図 2 は斜め上の視点から見た図、図 3 真横から見た図である。図 4 はファイルの情報を表示した図である。ファイル情報を取得したいファイルのアイコンをクリックすることにより、情報が画面にむかって左下に表示される。また、クリックしたアイコンは灰色で表示される。ディレクトリの場合においても同様である。各図とも表 2 で設定した表示形式に沿って表示されている。なお、表 3 における平均計算時間は約 0.15 秒であった。

図 5~7 は不正アクセスがあったファイルのアイコンを真上 (図 5)、斜め上 (図 6)、真横 (図 7) から見た図である。真上から見た図 5 において不正アクセスを表す球が消えてしまっている。これは OpenGL の Z バッファ精度の限界に起因するもので、視点をより近づければ球が見えるようになる。ファイルを表すアイコンと球の距離をより長く設定すればこの段階でも消えずに見えるが、真横から見た場合の違和感を考慮し、現段階ではこの状態としている。

1,000 個から 10,000 個のファイル数のデータ (1,000 個のデータを必要回数コピー) に対する提案手法の処理時間を表 4 に示す。それぞれ 10 回処理時間を計測し、その平均値を示している。提案手法内部では表示順位ならびに配置を決定するためのソーティングが含まれているが、主要処理は各ファイルに対する配置 (表示) である。このため、データ数と処理時間は概ね比例すると考えられ、表 4 からそれが裏付けられる。

表 2: 表示形式

対象	表現方法 設定
背景	R,G,B 255,255,255
円の内側と外側の枠	R,G,B 102,0,255
ディレクトリを区切る枠	R,G,B 0,0,0
アクセス制限ディレクトリ	R,G,B 255,0,0
不正アクセスファイル	R,G,B/標識 204,153,51/球
ファイル拡張子: gif	R,G,B 255,102,153
ファイル拡張子: html/htm	R,G,B 51,255,153
ファイル拡張子: jpg	R,G,B 51,0,153
ファイル拡張子: その他	R,G,B 128,128,128
アクセス元: ac.jp	形状 三角
アクセス元: go.jp	形状 丸
アクセス元: その他	形状 四角
順位	標識 矢印
アクセス数	形状 高さ
ディレクトリ数	形状 区切り線の数
ファイル数	形状 アイコンの数
ファイル/ディレクトリ情報	R,G,B 51,51,51

表 3: Web データ

全ファイル数	計 326
最上位階層ディレクトリに属さない ファイル数	計 21
ディレクトリ名 : img	計 32
ディレクトリ名 : product1	計 57
ディレクトリ名 : product1/member1	計 11
ディレクトリ名 : product2	計 57
ディレクトリ名 : product2/member2	計 11
ディレクトリ名 : service1	計 90
ディレクトリ名 : service2	計 69

4 結論

本稿では、大規模階層構造データが持つ識別情報を利用してデータの表示順序・表示形式を変更し、一面表示する手法を提案した。本稿では Web ページのアクセス数などを識別情報として想定し、対話型環境を考慮して情報の最適配置よりも処理時間を重視している。提案手法ではほぼリアルタイムでユーザが指定した順位により対象データを表示可能である。また、色や形の変化で、階層構造以外の多くの識別情報をも一面表示し、ユーザの理解・認識促進の一助となるものと考えられる。なお、提案手法は処理時間短縮を優先したため最適配置を行っていない。このため、一画面表示に重要な要素である表示の効率性についての検討・改良が今後の課題である。また、識別情報による表示変化がより一層認識しやすくなる表示方法についても今後の検討が必要である。

謝辞

本研究の一部は、文部科学省 21 世紀 COE プログラム「電子社会の信頼性向上と情報セキュリティ」のもと行われた。

参考文献

- [1] 伊藤貴之, 山口裕美, “「データ宝宝箱」～ビジュアルデータマイニングの実現に向けて～”, 情報処理学会 情報基礎学研究報告会 No.071,2003
- [2] Itoh T., Koyamada K., “HeiankyoView: Orthogonal Representation of Large-scale Hierarchical Data”, International Symposium on Towards Peta-Bit Ultra Networks (PBit 2003)

表 4: 処理時間

全ファイル数	時間/秒
1000	約 0.33
2000	約 0.58
3000	約 0.88
4000	約 1.53
5000	約 1.61
6000	約 1.85
7000	約 2.10
8000	約 2.49
9000	約 2.73
10000	約 3.01

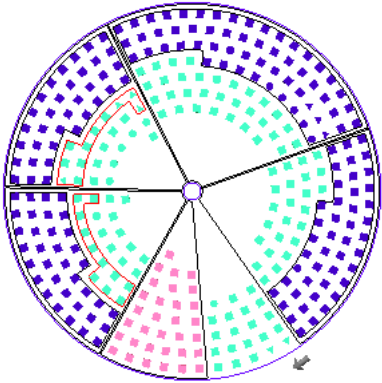


図 1: 視点: 真上

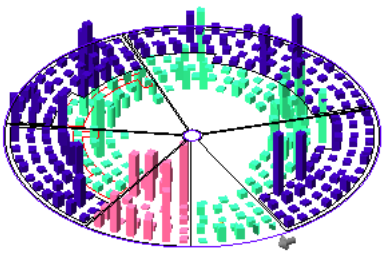


図 2: 視点: 斜め上

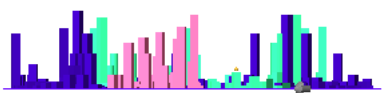


図 3: 視点: 真横

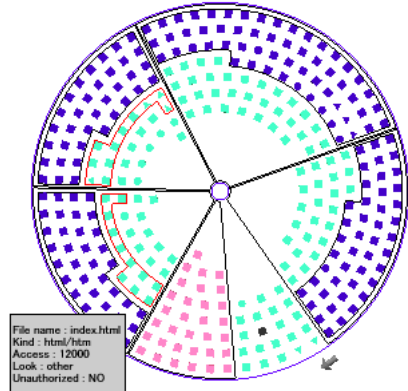


図 4: ファイル情報表示



図 5: 拡大表示・視点: 真上

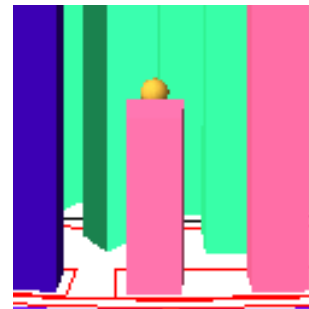


図 6: 拡大表示・視点: 斜め上

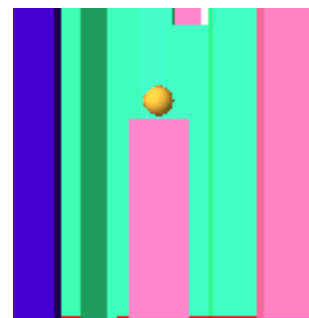


図 7: 拡大表示・視点: 真横