

ズームインタフェースを用いたグラフ階層閲覧のための一描画手法

A Method for Drawing Hierarchical Graph for Zooming Navigation Interface

豊田 正史*

Summary. 大規模な有向グラフを階層的にクラスタリングして得られるグラフ階層を、入れ子構造として可視化しズームインタフェースを用いて自在に閲覧可能にするためのグラフ描画の一手法を提案する。グラフ階層の可視化閲覧についてはすでに様々な手法が提案されているが、本手法ではノードの重なり回避、および階層間のエッジの距離を考慮した、単純かつ効果的な方法を示している。本手法を大規模なブログデータから抽出したグラフに適用し、話題の中心となっているノードをズームしながら閲覧できることを示す。また、1万ノード規模のグラフでも問題なく動作することを確認した。

1 はじめに

Web グラフ、ソーシャルネットワーク、生物学的ネットワークなど、複雑なネットワーク構造分析が盛んに行われているが、様々な分析から得られる膨大な結果を理解するためには、大規模なグラフの可視化および閲覧システムが必要不可欠である。大規模グラフの閲覧には、ノードを階層的にクラスタリングして可視化する手法が有効であり、近年提案された代表的な手法としては、Treemap と Fisheye view を併用した手法 [1]、ツリービューと選択した階層のグラフ描画を併用した ASK-GraphView [2]、円形の入れ子構造を用いた Grouse [3]、GrouseFlocks [4] などがあげられる。

本論文では、大規模な有向グラフを階層的にクラスタリングして得られるグラフ階層を、Grouse と同様に円形の入れ子構造として可視化し、ズームインタフェースを用いて自在に閲覧可能にするためのグラフ描画の一手法を提案する。本手法では、リアルタイムにレイアウトをしつつ、ノードの重なり回避、および階層間のエッジの重なり回避を行うための、従来手法より単純で効果的な方法を示している。本手法を大規模な RSS フィードデータから抽出したグラフに適用し、話題の中心となっているノードをズームしながら閲覧できることを示す。また、1万ノード規模のグラフでも問題なく動作することを確認した。

2 グラフ階層描画手法

提案するグラフ階層描画手法は、入力を有向グラフ $G = (V, E)$ とする。ノード集合 V のクラスタリング手法としては、 V 互いに素な部分集合に分割するものを想定している。例えば、連結成分分解、強連結成分分解、1-連結成分分解、k-core、Newman

法など様々な手法が存在する。分割 P は、以下のよう表せる。

$$P = \{P_k | P_k \subseteq V, P_i \cap P_j = \emptyset (i \neq j)\}$$

階層的なクラスタリングとは、分割を個々の P_k に対して再帰的に行うことを指し、結果は V を根ノードとする木構造となる。すなわち、各ノード $v \in V$ は、それを含む唯一の分割を親ノードとして持ち、根ノード以外の親ノードも、それを含む唯一の分割を親ノードとして持つ。図 1 に実装したシステムの画面スナップショットを示す。本システムでは、グラフ階層が円形の入れ子として表現され、ズームインしていくことで、深い階層の詳細を閲覧することが可能となっている。図 1 では、上から順に階層にズームインしていく様子を示している。図 1 は、約百万ブログのフィードを 3 年間にわたり毎日収集したアーカイブから、「ワーキングプア」について言及したブログを抽出し、その間のリンク関係を示したものである、ノード数の大きく密にエッジの張られたクラスタにズームをしていくと、話題の中心に NHK の特集番組があることが見て取れる。

本システムでは、デフォルトのグラフの階層的クラスタリング手法として、peeling および k-core という既存の手法を用いている。Peeling は、次数が 1 のノードを、それが接続している唯一のノードに縮約することを繰り返す手法であり、木構造を持つ部分グラフをまとめる事ができる。k-core は、グラフ G の部分グラフであり、含まれる全ノードの次数がその部分グラフ中において k 以上のものとして定義される。k-core は、元のグラフから、次数 k 未満のノードを繰り返し削除していくことで得られ、 $(k+1)$ -core は、必ず 1 つの k-core に含まれるという特徴を持つ事から、階層的なクラスタリング手法として用いる事ができる。個々の k-core は連結でない可能性があるが、その場合にはその中の各連結成分を 1 つのクラスタとする。

各親ノードはそれぞれ独立した座標系を持ち，その上で子ノードのレイアウトを行う．Grouse[3]と同様に，各親ノードにおいて子ノードのレイアウト手法を選択することが可能である．現在の実装では，力学的モデルに基づいたレイアウト手法として，Eadesのspring-embedder [5]，Fruchterman，Reingoldによるforce-directed placement [6]などを選択することができる．

各ノードは大きさを持つため，重なりを回避することが必要となる．重なり回避の様々な手法が提案されているが，事後のレイアウト修正を行う必要のある手法が多い．本手法では，斥力の最大値を設定し，ノード同士が重なっている間は，常に最大の斥力が働くように斥力を修正し，引力および斥力のパラメタであるノード間の距離についてもノードの大きさを反映することで重なり回避を行っている．

また，力学的レイアウトに用いられる力に加え，階層間のエッジに関する力を子ノードに直接加えることにより階層間エッジをできるだけ短くレイアウトするヒューリスティックを提案する．異なる階層間では，スケールが異なるため，加える力は距離によって決めるのではなく，回転によって与えることとする．各親ノードのレイアウトを崩さず，親ノード全体を回転させる方法が提案されているが，本手法では，各子ノードに直接回転を与える点が異なり，原理的には階層間エッジの長さをより短くすることが可能である．

本手法は，Javaを用いて実装されている．Intel Core2 Duo(2.6GHz)，Quadro FX 360M ビデオカードを搭載したマシンで動作を確認しており，1万ノード規模のグラフでも，20fpsでストレスなく動作している．

参考文献

- [1] J. Abello, S. G. Kobourov, and R. Yusuf. Visualizing Large Graphs with Compound-Fisheye Views and Treemaps. In *Graph Drawing*, pp. 431–441, 2004.
- [2] J. Abello, F. van Ham, and N. Krishnan. ASK-GraphView: A Large Scale Graph Visualization System. *IEEE Trans. Vis. Comput. Graph.*, 12(5):669–676, 2006.
- [3] D. Archambault, T. Munzner, and D. Auber. Grouse: Feature-Based, Steerable Graph Hierarchy Exploration. In *EuroVis*, pp. 67–74, 2007.
- [4] D. Archambault, T. Munzner, and D. Auber. GrouseFlocks: Steerable Exploration of Graph Hierarchy Space. *IEEE Trans. Vis. Comput. Graph.*, 14(4):900–913, 2008.
- [5] P. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 42, pp. 149–160, 1984.
- [6] T. M. J. Fruchterman and E. M. Reingold. Graph Drawing by Force-directed Placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

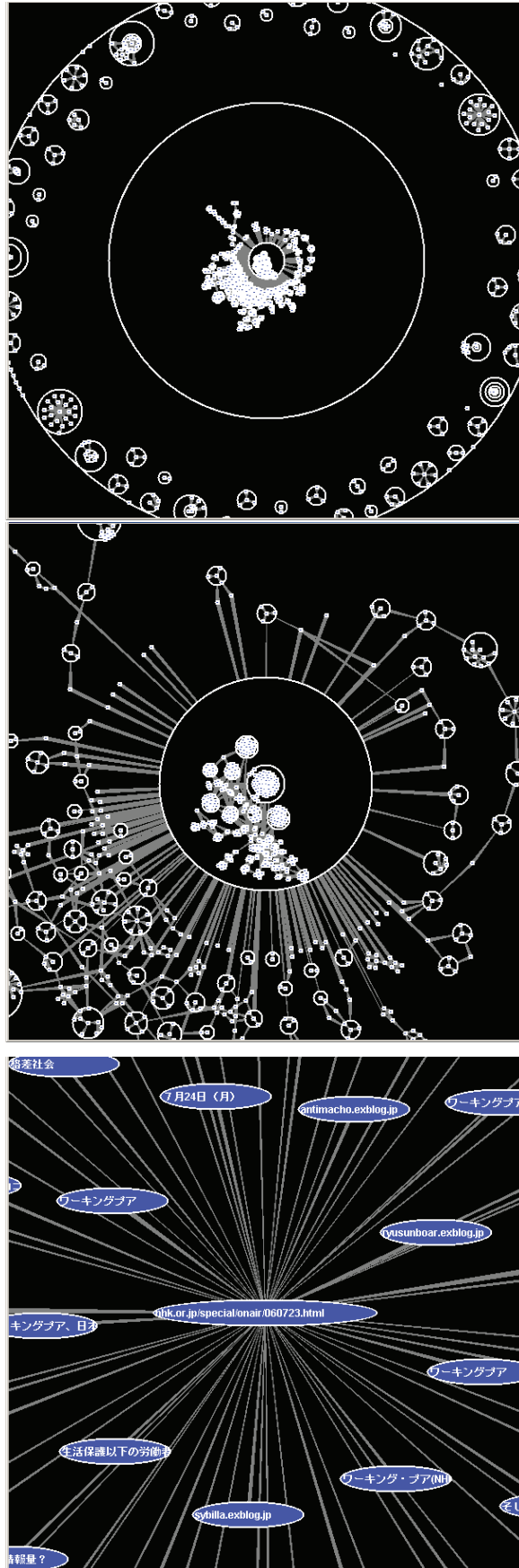


図 1. グラフ階層の可視化例