Focus Dependent Multi-level Graph Clustering

François Boutin
PhD Student
LIRMM - UMR 5506 du CNRS
Université Montpellier II – France
francois.boutin@univ-montp1.fr

Mountaz Hascoët
Associate Professor
LIRMM - UMR 5506 du CNRS
Université Montpellier II – France
mountaz@lirmm.fr

ABSTRACT
In this paper we propose a structure-based clustering technique that transforms a given graph into a specific double tree structure called *multi-level outline tree*. Each meta-node of the tree – that represents a subset of nodes – is itself hierarchically clustered. So, a meta-node is considered as a tree root of included clusters.

The main originality of our approach is to account for the user focus in the clustering process to provide views from different perspectives. *Multi-level outline trees* are computed in linear time and easy to explore. We think that our technique is well suited to investigate various graphs like Web graphs or citation graphs.

Categories and Subject Descriptors
H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering*.

H5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Hypertext navigation and maps*.

General Term
Algorithms

Keywords
Graph Drawing, Graph Clustering, multi-scale visualization

1. INTRODUCTION
The idea of structure-based clustering is to minimize the number of links between clusters and maximize intra-connectivity [7]. Hierarchical clustering proceeds iteratively with merging or splitting the most appropriate clusters according to distance between clusters [3][4]. Whereas partitioning techniques are based on iterative optimization algorithms [1]. A partitioning method is usually used because of its efficiency and a hierarchical clustering is used because of its quality [10].

We presented previously a link-based focus-dependent clustering technique that provides a "flat" *outline tree* [2]. Now we propose a multi-level clustering method that transforms an undirected graph in a specific structure called *multi-level outline tree*.

AVI '04, May 25-28, 2004, Gallipoli (LE), Italy

It is a double-tree based on an adjacency tree and an inclusion tree. This structure presents two advantages: (1) It is displayed without edge crossing, (2) Views depend on user focus. So we think that visualization and interaction will be more effective.

We first introduce multi-level graph clustering concepts. Next we describe our multi-level clustering algorithm. Then, we present a citation graph exploration based on a *multi-level outline tree*.

2. Multi-level structures - definitions
A graph G is defined by a set of vertices $V=\{v_i, 1\leq i \leq N\}$ and a set of edges $E=\{(v_i,v_j), 1 \leq i < j \leq N\}$. We denote G=(V,E).

We define two multi-level structures used below:

- A hierarchical clustered graph is defined by a graph G = (V,E) and a rooted tree T [4]. Leaves of T are vertices of G. Each node of T is a set of nodes of G called cluster. T describes an inclusion relation between clusters so T is called inclusion tree [3]. We present (figure 1) a hierarchical clustered graph where T is presented by an inclusion of circles.
- A compound graph consists of a hierarchical clustered graph with edges between clusters [8][9]. We show (figure 2) a 3D view of a compound graph [4]. Arrows represent edges of inclusion tree T. Dotted lines represent edges between clusters. Their removal provides a 3D hierarchical clustered view.

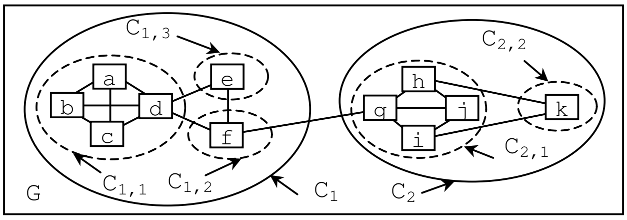

Figure 1: multi-level graph clustering

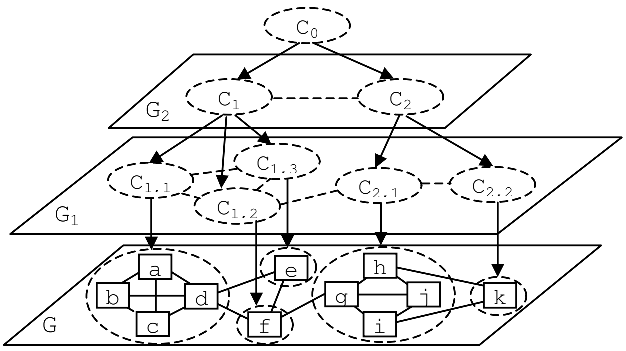

Figure 2: Compound graph

3. Focus-Based Multi-Level Clustering

3.1 Preliminary definitions

We apply our technique to an undirected connected graph G=(V,E). We consider v_1 as a specific vertex called *focus*.

- The *depth* of node v_i denoted by $d(v_i)$ is the distance between v_i and v_1 (the length of the shortest path between v_1 and v_i).
- Depth of G is defined by $d_{max} = \max \{d(v_i), v_i \in V\}$.
- L_m (*m-layer*) is defined by $L_m = \{v_i \in V, d(v_i) = m\}$.
- $G^k{}_m$ is defined by $G^k{}_m = \{v_i \in V, m \leq d(v_i) \leq m+k\}$.
- $G^\infty{}_m$ is the set defined by $G^\infty{}_m = \{v_i \in V, d(v_i) \geq m\}$.
- $C^k{}_{m,j}$ is the j^{th} connected component of $G^k{}_m$.
- $C^\infty{}_{m,j}$ is the j^{th} connected component of $G^\infty{}_m$.

3.2 k-relative relation and k-clusters

- Definition: Nodes v_i and v_j on a m-layer L_m are said *k-relatives* if there is a path between v_i and v_j in $G^k{}_m$. They are said *relatives* if there is a path in $G^\infty{}_m$.
- Definition: k-relative-relation is an equivalence relation on L_m. Consequently for a m-layer L_m and a given k, we get equivalence classes called *k-clusters*. In practice a k-cluster is defined by $V^k{}_{m,j} = \{v, v \in L_m \cap C^k{}_{m,j}\}$. Similarly on a m-layer L_m we define *meta-nodes* by: $V^\infty{}_{m,j} = \{v, v \in L_m \cap C^\infty{}_{m,j}\}$.

For instance (figure 3a) nodes 3 and 4 are 0-relatives and belong to $V^0{}_{2,2}$. Nodes 5 and 8 are 1-relatives and belong to $V^1{}_{3,1}$ but they are not 0-relatives. Nodes 2 and 3 are 2-relatives since the path (2, 6, 15, 9, 8, 3) belongs to $G^2{}_2$ but they are not 1-relative.

3.3 Inclusion tree of k-clusters

- Property: On a m-layer L_m, if v_i and v_j are k-relatives, they are (k+1)-relatives and relatives. So a k-cluster is included in a (k+1)-cluster and belongs to the corresponding meta-node.
- Property: Each meta-node is the root of an *inclusion tree* whose nodes are k-clusters. Leaves are 0-clusters.
- Definition: G contains all meta-nodes. So it can be considered as the root of an *inclusion tree* of k-clusters.
- Definition: *Singleton-clusters* are k-clusters that contain only one (k-1)-cluster. To avoid cluttering views, singleton-clusters are not displayed in the final view.

For instance (figure 3a) nodes 3 and 4 belong to $V^0{}_{2,2}$ but also to $V^1{}_{2,2}$ and $V^2{}_{2,1} = V^\infty{}_{2,1}$ (meta-node). Note that $V^1{}_{2,2}$ is not represented (in figure 3e) since it is a singleton-cluster.

3.4 Adjacency tree of meta-nodes

- Definition: Two k-clusters are said to be *linked* if they contain nodes that are linked (see quotient graph **[3]**). We present figure 3, for k = 0, 1 and 2 the graph made of k-clusters and links between them.
- Property: Meta-nodes belong to an *adjacency tree* called *outline tree* since it looks like the outline of G [2].

3.5 Multi-Level Outline Tree

- Definition: Graph made of both inclusion tree and adjacency tree is called *multi-level outline tree* (see figure 3e).
- Property and definition: A *multi-level outline tree* is a compound graph (see **[8][9]**) based on an adjacency tree. So we call it a *compound tree*.

We show (figure 3e) a *multi-level outline tree* based on focus 1.

4. Computing

Three steps are necessary to compute a *multi-level outline tree* in linear time:

- Step 1: To begin with we treat 0-clusters. For each m-layer L_m we have to identify connected components that represent 0-relative-components called 0-clusters (figure 3b). Then we consider an edge between two 0-clusters if they contain two connected nodes. For instance (figure 3b) $V^0{}_{2,2}$ and $V^0{}_{3,2}$ are two adjacent 0-clusters.
- Step 2: We build iteratively k-clusters from (k-1)-clusters (for $k \geq 0$) using 1-relative-relation instead of k-relative-relation. Thus, a *multi-level outline tree* is easier to compute: If two (k-1)-clusters denoted $V^{k-1}{}_{m,i}$ and $V^{k-1}{}_{m,j}$ are 1-relatives, we build a k-cluster denoted $V^k{}_{m,r}$ that contains these two (k-1)-clusters. We prove easily that it produces the same clusters when we use k-relative-relation on G. For instance (figure 3) $V^1{}_{2,1}$ and $V^1{}_{2,2}$ are 1-relatives because they are connected to $V^1{}_{3,1}$. So we build $V^2{}_{2,1}$ that contains $V^1{}_{2,1}$ and $V^1{}_{2,2}$.
- Step 3: The multi-level clustering algorithm stops when for a (k+1)-level we have only singleton clusters. It means that all (k+1)-clusters contain only one k-cluster. The resulting graph is a tree called *multi-level outline tree* (see figure 3e).

5. Results

We collected articles on bibliographic site www.citeseer.com [6] from a focus paper: "*Navigation and Interaction in Graphical Bookmarks*". We got recursively 4 levels of related papers from the focus using "*Active bibliography (Related documents)*" links. We built a graph with 61 nodes (papers) and links between them.

- A classical spring layout of this graph is presented figure 4. Note that the title of the paper is not displayed to avoid a cluttered view.
- A *multi-level outline tree* is computed from focus 1 (figure 5). It is easy to display a part of the title.

Additional visual tips have been added: the more a node has connections, the darker its background is. Similarly, the stronger k-clusters elements are connected the darker their background is. For instance, nodes 22, 23, 50, 51, 24 are strongly connected.

Interaction is used to display information dynamically. Node's relations become visible, when a user's pointer comes over the node. At the same time, all connected nodes are also highlighted and a tool-tip displays the entire title of the associated paper.

User can change the focus by simply clicking on a node. The *multi-level outline tree* is then re-computed (figure 6: focus has been changed to node 25).

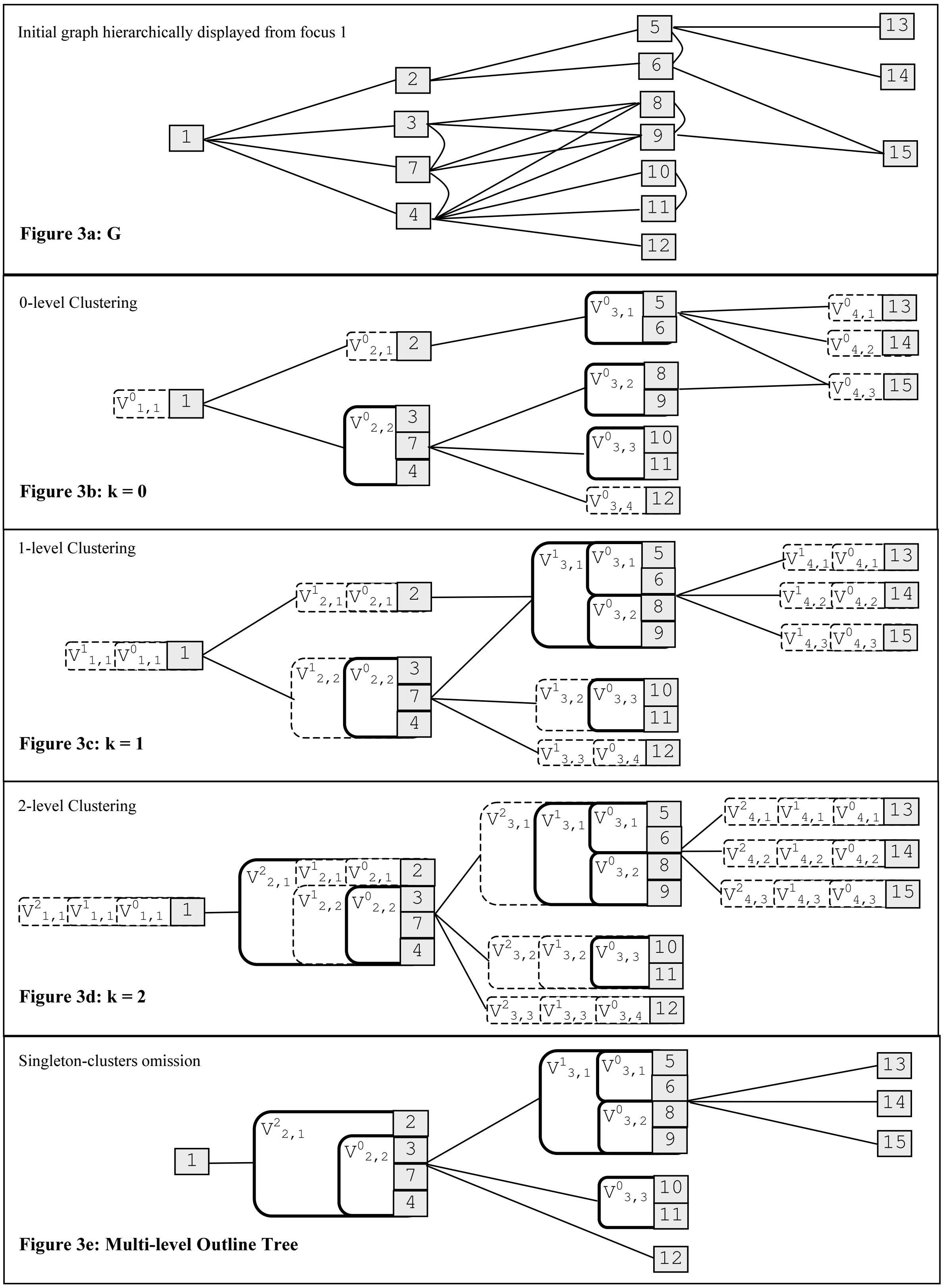

Figure 3: ***Multi-level outline tree*** **– step by step**

6. Conclusion

We described a multi-level clustering method based on a focus that provides a *multi-level outline tree* in linear time. It presents a double tree structure called *compound tree*. The graph is displayed as an adjacency tree of meta-nodes. Each meta-node is the root of an inclusion tree of clusters. Thus we can zoom hierarchically on a meta-node to access different levels of graph details.

More generally most multi-scale interaction techniques presented in [5] can be reused in order to provide an adequate interaction model.

Our technique is focus-dependent. So we can get different views of the graph depending on the perspective the user takes.

We believe that our clustering method is general enough to be applied to different fields: Web navigation, bibliography, social graphs.

7. Bibliography

[1] C. J. Alpert and A. B. Kahng, "Recent Directions in Netlist Partitioning: A Survey", INTEGRATION: the VLSI journal, vol. 19, pp.1-81,1995.

[2] F. Boutin & M. Hascoët (2003). "Focus-Based Clustering for Multi-Scale Visualization". Proceedings of the Conference on Information Visualization, IV'2003, IEEE, pp 53-59.

[3] R. Brockenauer, S. Cornelsen, "Drawing Clusters and Hierarchies". In Michael Vaufmann and Dorothea Wagner (Eds.): Drawing Graphs: Methods and Models, LNCS 2025, pp. 194 - 228. © Springer-Verlag, 2001.

[4] P. Eades, "Multilevel Visualization of Clustered Graphs," Proceedings of Graph Drawing'96, Berkeley, California, September,1996.

[5] M. Hascoët, "Interaction and visualisation supporting web browsing patterns". Information Visualization, IV'2001, London, IEEE, p 413-419, 2001.

[6] S. Lawrence, K. Bollacker, and C. L. Giles. Indexing and retrieval of scientific literature. In Eighth International Conference on Information and Knowledge Management, CIKM 99, pages 139-146, Kansas City, Missouri, 1999.

[7] T. Roxborough and A. Sen, "Graph clustering using multiway ratio cut," Proceedings of Graph Drawing Symposium, GD'97 Rome, September 1997. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 1353, pp. 291-296, 1998.

[8] G. Sander, "Layout of Compound Directed Graphs" Technical Report A03-96, Universität des Saarlandes, FB 14 Informatik, 1996

[9] V. Sugiyama & V. Misue, "Visualization of structural information: automatic drawing of compound graphs" IEEE Trans. System Man Cybernetics 21 (1991) 876-892

[10] M. Steinbach, G. Karypis, V. Kumar, "A Comparison of Document Clustering Techniques", Proc. TextMining Workshop, KDD 2000.

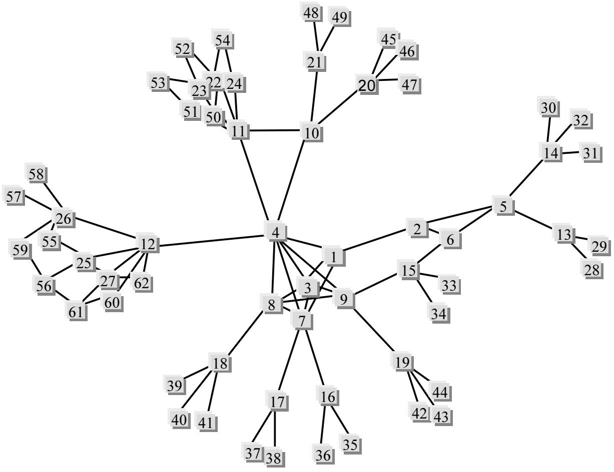

Figure 4: Graph G – a spring view – *invariant sets*

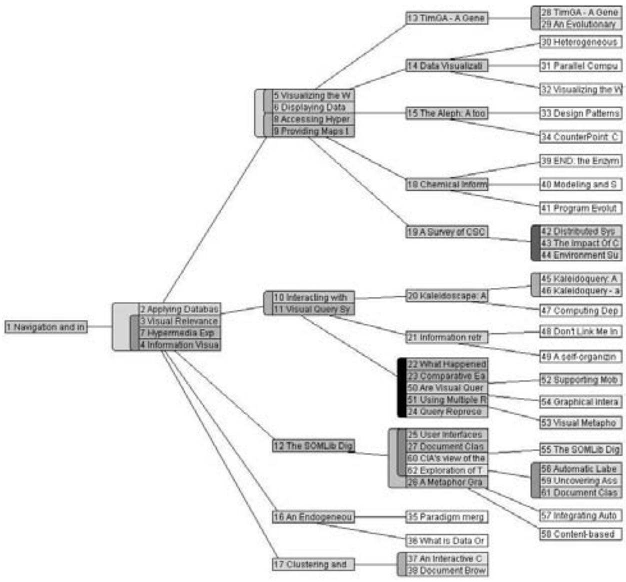

Figure 5: Multi-level outline tree from focus 1

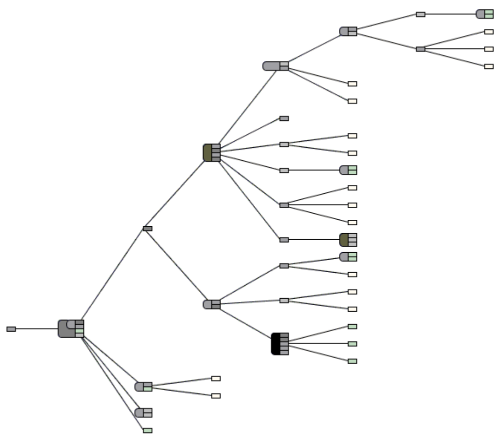

Figure 6: Multi level outline tree from focus 25