Visual Analysis of Graphs with Multiple Connected Components

Tatiana von Landesberger*
Interactive Graphics Systems Group,
TU Darmstadt and Fraunhofer IGD, Germany

Melanie Görner†
Interactive Graphics Systems Group
TU Darmstadt, Germany

Tobias Schreck‡
Interactive Graphics Systems Group
TU Darmstadt, Germany

ABSTRACT

In this paper, we present a system for the interactive visualization and exploration of graphs with many weakly connected components. The visualization of large graphs has recently received much research attention. However, specific systems for visual analysis of graph data sets consisting of many such components are rare. In our approach, we rely on graph clustering using an extensive set of topology descriptors. Specifically, we use the Self-Organizing-Map algorithm in conjunction with a user-adaptable combination of graph features for clustering of graphs. It offers insight into the overall structure of the data set. The clustering output is presented in a grid containing clusters of the connected components of the input graph. Interactive feature selection and task-tailored data views allow the exploration of the whole graph space. The system provides also tools for assessment and display of cluster quality. We demonstrate the usefulness of our system by application to a shareholder structure analysis problem based on a large real-world data set. While so far our approach is applied to weighted directed graphs only, it can be used for various graph types.

Index Terms: E.1 [Data Structures]: Graphs and Networks—[H.3.3]: Information Search and Retrieval—Clustering H.5.2 [User Interfaces]: Graphical user interfaces (GUI)— [I.3.3]: COMPUTER GRAPHICS—Picture/Image Generation

1 INTRODUCTION

Visual exploration of graphs is needed in various application areas, e.g., corporate governance, supply chain management, networks of biologic reactions, cash-flow analysis, etc. The subsets of entities connected to each other within the whole network create many weakly connected components (further referred to also as connected components or components). For example, in shareholding network, each component represents one company holding "structure". When analyzing the whole economy, often the exploration of the network with regard to distribution of components according to various criteria is needed. For example, *Are there corporate holdings typical to the studied economy?* or *are there differences in the shareholding relationships between different sectors in a given economy?*. Interactive visual exploration systems can help analyzing such networks.

The visualization of large graphs (with tens of thousands of nodes or more) has recently received much research attention [2, 14, 19], and results of it have been applied to visual analysis of large network data sets from different domains. However, only few specialized approaches are available for visual exploration of graphs with multiple components [17, 20]. They usually use packing algorithms for non-overlapping and space efficient layout of graph components. These approaches, however, disregard similarities between the components.

*e-mail: tatiana.von.landesberger@gris.tu-darmstadt.de
†e-mail: mgoerner@gris.tu-darmstadt.de
‡e-mail: tschreck@gris.informatik.tu-darmstadt.de

In this paper, we present a novel system for the interactive visualization and exploration of graphs with many connected components. Our approach follows Keim's visual analytics Matra "Analyse First – Show the Important – Zoom, Filter and Analyze Further – Details on Demand" [31]. We use clustering of connected components for gaining overview of the prototype subgraphs in the network. The Self-Organizing Map (SOM) algorithm is used as it offers robust clustering and is well suited for visualization. The SOM cluster analysis is driven by a rich set of topology-based graph features, which can be interactively selected and combined by the user. Depending on the feature selection, different insights into the dataset can be produced. Our system offers exploration of the graph space by interactive visualization of the clustering results and clustering quality. Furthermore it offers the possibility of saving and loading of clustering results using different clustering parameters and user-defined annotations of the process. It supports reproducibility of the analysis and insights gained.

The remainder of this paper is structured as follows: Section 2 reviews related work on visualization of large graphs and of disconnected graphs. In Section 3, we introduce the overall architecture of our proposed visual graph analysis system. Section 4 recalls the SOM clustering procedure, and introduces our set of graph features we use for clustering. Section 5 describes the main visualization and interaction concepts proposed for exploration of the disconnected graph space. Section 6 discusses an application of the technique to German shareholder network. Finally, Section 7 concludes and discusses options for future work.

2 RELATED WORK

The visualization of large graphs is one of the main research areas in information visualization. In this section, we introduce some of the layout algorithms for large graphs and then concentrate on visualization of graphs with multiple components. Approaches applying SOM in connection with graph visualization are also discussed.

In the following we recall several layout algorithms, however our goal is not to elaborate on the wide variety of approaches. A general overview of *graph visualization techniques* can be found in [24]. For visualization of large general graphs many different approaches have been used, e.g., [19, 27, 1, 35, 23, 18, 2]. They often include the assessment of graph topology (by graph features) for a better layout of the graphs. Large graph visualizations often use data reduction techniques such as clustering [9, 34], filtering [37], multi-scaling [3] or edge bundling [25, 14]. *Grouping of components* has been addressed by Cohen and Deschamps [13]. They use minimal paths between components for finding complete curves from a set of edge points for image analysis. *The visualization of graphs with multiple components* uses "packing", i.e., it lays out the components so that they do not overlap and are space efficient. For example, Dogrusoz [15] compares various two-dimensional packing algorithms for graphs which use the representation of graphs by their bounding rectangles. They include strip packing, tiling and alternate-bisection. The polyomino algorithm [17] of Freivalds et al. uses polyomino representation of the graph objects which substantially reduces the unused display space in comparison to rectangular shapes. Goehlsdorf et al. [20] introduce new quality measures to evaluate a two-dimensional placement which yields more com-

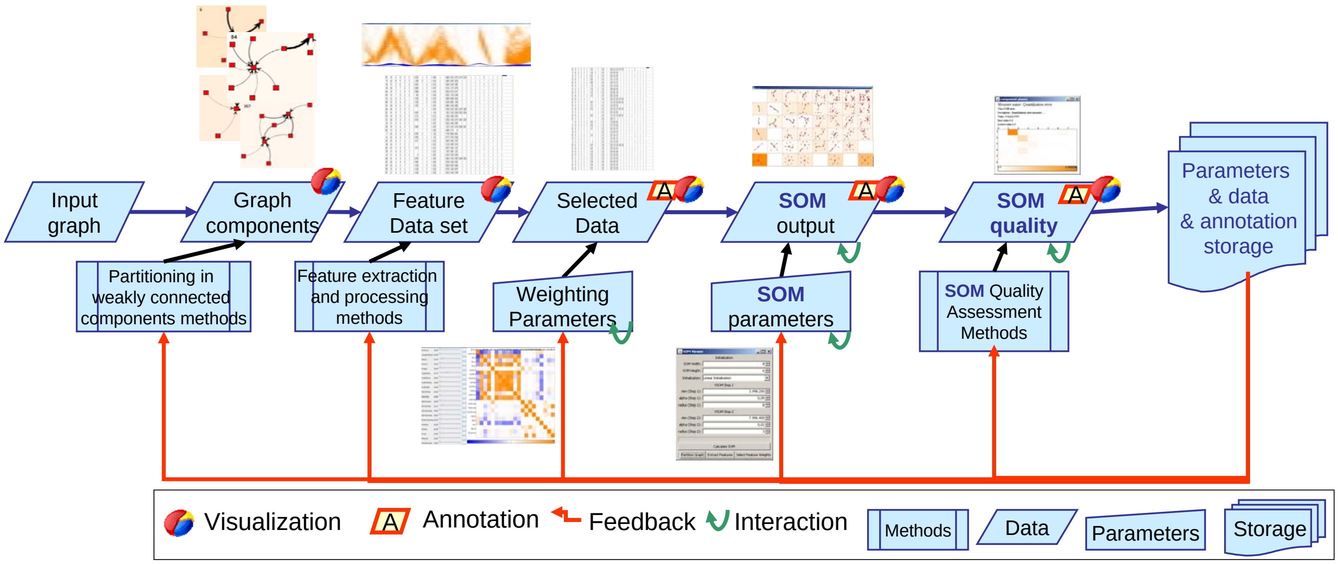


Figure 1: Architecture model of our system for visual analysis of graphs with multiple components.

pact layouts than the previously mentioned approaches.

The usage of self-organizing maps for graph drawing has been introduced by Meyer and Bonabeau. Meyer [39] described the so called ISOM layout, which is an extension of self organization strategies for drawing undirected graphs. Bonabeau applied SOM to laying out large graphs in 2D [7] and to multi-dimensional scaling of large graphs [6]. However, all these approaches are primarily developed for *visualizing connected graphs*. Prieto et al. [44] use SOM for the visualization of the evolution of a web-based social network. According to the authors, owing to the simple graph description using the graph adjacency matrix, their approach is *constrained to graphs with a maximum of 500 edges*. Their focus is the visualization of time-variation of the network. SOM-based clustering for graph matching has been used by Gunter and Bunke [21]. They use edit-based graph distance for the recognition of handwriting. However, this approach has not been applied to general graphs and moreover, their work does not include visualization of clustering results. Neuhaus and Bunke [40] also uses SOM with graph edit cost for graph matching applying the approach to numerically labeled graphs.

In comparison to the approaches presented above, we offer a clustering of connected components according to their topologic properties and interactively explore the results.

3 Interactive Graph Clustering Process

Clustering is an important data analysis technique. It supports the examination of large amounts of data by abstraction to a limited number of data prototypes describing groups of data and providing overview of the whole dataset. We now present our system for visual analysis of graphs with multiple connected components using SOM clustering based on feature description of the components.

The clustering-based graph analysis process (cf. Figure 1) starts with partitioning the whole input graph into its weakly connected components. Each extracted component is described by a set of topologic features creating a feature vector data set (cf. Section 4.2.1). The summary table of the feature data set or the whole feature vector data set can be explored using various views (e.g., using multivariate visualization techniques such as parallel coordinates). The following feature selection step is supported by a user interface for interactive weight adjustment (cf. Section 4.2.2). The selected feature set is used as input for calculating similarity between components during clustering. After choosing the SOM clustering parameters, SOM clustering is performed (cf. Section 4.1). The clustering results are shown using interactive visualization techniques described in Section 5.1. The subsequent assessment of the clustering quality is supported by interactive exploration of the results as well as by calculation of the SOM quality measures and their detailed display (cf. Section 5.2). During the whole process, the system supports to add user annotations to each analytic step. These annotations can be used to capture parameter decisions made, or intermediate analysis results obtained while working with the system. The results of the process (input and output data, annotations and parameters) are stored, supporting reproducibility and comparison of results. The feedback loop allows to change the parameters, switch between process stages and views and thereby create new results and insights.

4 Graph Features and Graph Clustering

In this section our approach to clustering of large disconnected graphs is described. As a starting point, graphs with large amounts of weakly connected components are assumed. Each component is described by a set of features which are used to determine the similarity between graphs. We employ the well-known Self-Organizing-Map algorithm as the clustering technique of choice. The SOM method is described in Section 4.1, while features used are introduced in Section 4.2.

4.1 Clustering Approaches and Self-Organizing Maps

To date, many clustering techniques relying, e.g., on centroid or medoid-based approaches, hierarchical or density-based models, have been proposed [30, 28, 22]. While they differ in the way the clusters are obtained, most algorithms rely on an implementation of a *similarity* function defined over the set of data elements which are to be clustered. *Transformation-based* approaches such as the Edit-2 Distance for undirected acyclic graphs [48] calculate the distance between graphs as the sum of costs when efficiently transforming one graph into the other. *Feature-based* approaches such as the graph histogram technique [42] capture important data attributes in form of a vector or histogram. Consequently, distances between data elements are calculated using vector-space or histogram distance functions. In this paper, we employ the latter technique.

The Self-Organizing Map algorithm (SOM) [32] is a neural network learning algorithm with a strong disposition for visualization [47]. A network of prototype vectors is iteratively trained to represent a set of input data vectors in linear run time. The network is often assumed to be a 2-dimensional regular grid. During training, the algorithm iterates over the input data vectors. For each input vector, it finds the best matching prototype, and adjusts it as well as a number of its network neighbors toward the input vector. In the course of the process, the considered neighborhood size and the strength of the adjustment process (learning rate) are reduced. The training results in a set of prototype vectors (or clusters) representing the input data. In addition, the low-dimensional arrangement of prototypes on the network yields a topological ordering of the prototype vectors, approximating the topology of data samples in original data space.

SOM analysis has previously successfully been applied to many different data types including documents [26], audio [45], and images [5]. In this work, we apply SOM cluster analysis based on feature vector descriptions of the graphs in our data set. In our current implementation, we use the SOMPAK [33] engine together with typical rule-of-thumb training parameter settings recommended for initial SOM clustering [33]. Specifically, we use a $8*6$ rectangular SOM lattice with bubble neighborhood kernel. Other SOM lattice sizes and training parameters can be interactively user-specified.

4.2 Graph Features

For SOM analysis, we describe a graph structure by a set of appropriate graph topology properties (features). The selection of features, in general, depends on the type of network (directed vs. undirected, weighted vs. unweighted, with vs. without node labels, with vs. without node weights etc.) Moreover, the network semantic plays a role when defining the set of graph features to use. For instance, the sum of all weights on incoming links in shareholder networks should always sum up to 100% and is therefore not informative. In contrast, the same measure in flow networks illustrates the flow strength through the nodes.

4.2.1 Definition of Graph Features

Based on graph topology literature [11, 4, 8, 46, 12, 42] we chose a set of graph features referred to as important for weighted directed graphs. We categorized our supported features into general features, degree distribution features, distance features, reciprocity features, clustering features, and motif-based features. We briefly introduce each feature set in the following. For detailed definitions of features in each set, we refer to the above-mentioned literature.

1. ***General features*** measure general properties of a network. Examples include the size of a network (number of nodes), the degree of completeness (number of links relative to the number of possible links), the average edge weight.

2. ***Reciprocity features*** in a directed network indicate whether node links are reciprocal, meaning that if there is a link from nodes A to B, then there exists also a link from nodes B to A. The set includes (weighted) reciprocity and the correlation coefficient of the adjacency matrix.

3. ***Distance features*** measure lengths of paths between nodes in a network, e.g., diameter of a graph.

4. ***Clustering features*** measure the probability that two nodes that are neighbors to a third node, also share a link between each other. Different measures of clustering coefficients (weighted, in/out-clustering, etc.) can be used.

5. ***Degree distribution features*** show the division of nodes according to their (in/out)degree. The features include average/maximum relative node degree, relative number of loops, relative number of leaves, and relative number of roots. Additionally, degree correlation/assortativity can be used.

6. ***Motif-based features*** measure structural properties of graphs by the frequency of certain predefined substructures (motifs) occurring in the given graph. Analysis of many network structures, such as biological or company networks, is assisted by examining the occurrence of certain motifs. In particular, in shareholder structure analysis, these motifs allow to describe, e.g., companies with many subsidiaries (out-star motif), or structures leading to strengthening the voting power in a company via holding shares in third companies (feed-forward or caro motifs). Figure 2 shows the set of considered motifs.

The introduced features do not cover all possible graph types and all graph domains and therefore can be extended according to the particular graph type and the use case. For example, for labeled networks, features describing label distribution can also be used, or in other cases, centrality features may be relevant as well. All the above-mentioned features have their strengths and weaknesses for different analysis domains. The particular set will be determined by the given analysis task and can be selected interactively (see the following section).

We note that the time complexity of the feature extraction typically depends on number of connected components, the size of each component, and the given type of feature. The calculation often can be accelerated by using parallel processing techniques. As the feature calculation could be done offline and only once for each graph and feature, also more expensive features can be considered, given the available resources.

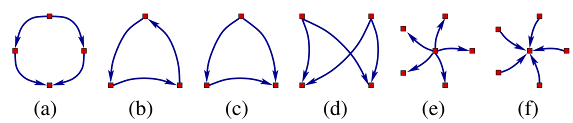


Figure 2: Selected graph motifs. a) Caro, b) Feedback, c) Feedforward, d) Bipartite, e) Out-star and f) In-star.

4.2.2 Normalization and Selection of Features

The selection of features, their normalization and weighting influences the result of the SOM clustering. Depending on the type of network, use case and user task at hand, an appropriate combination of graph features needs to be formed. We would not like to offer a predefined set of selected features (based on our experience), but

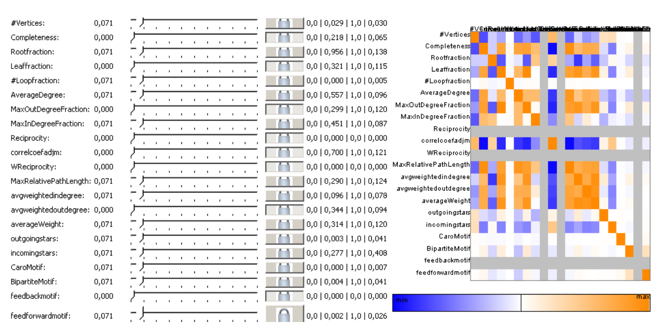


Figure 3: User interface for selection and weighting of features including a feature correlation matrix.

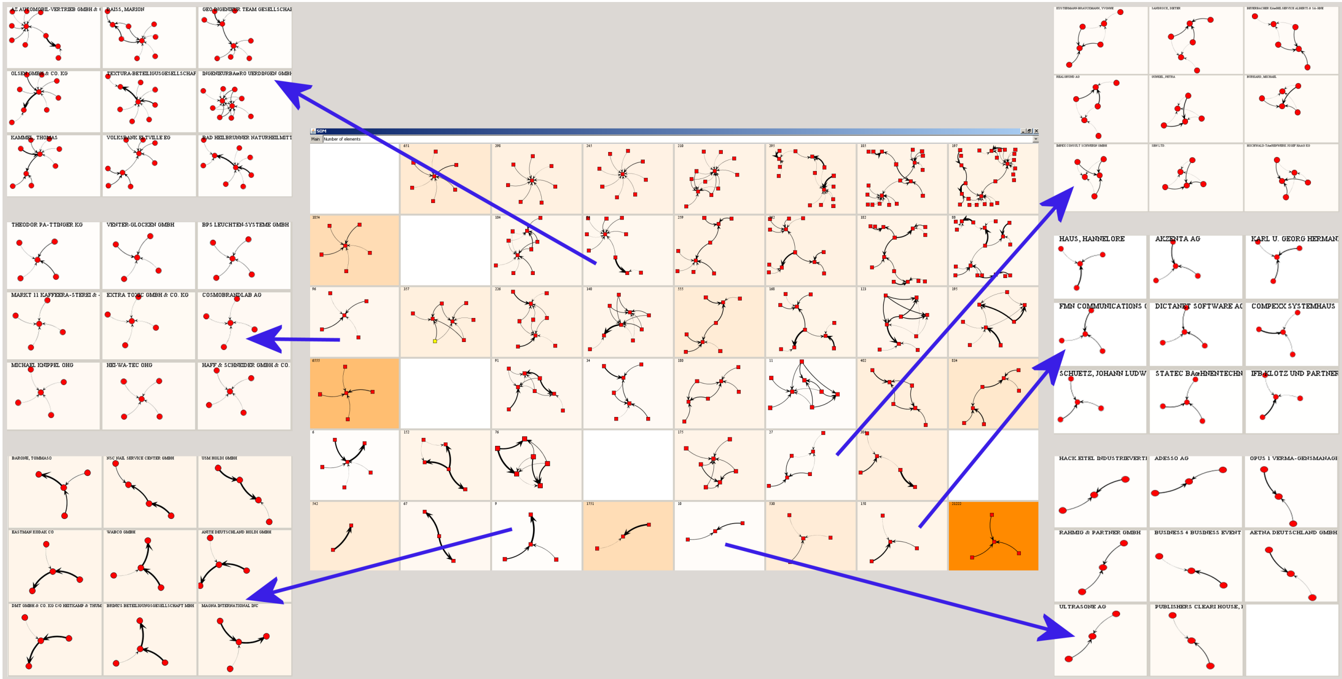


Figure 4: Visualization of SOM clustering results (center). Each cell contains the nearest neighbor graph, while the background color indicates the frequency of the cluster elements at each cluster. Along the cluster map, several member views showing a set of nearest cluster members is shown. The member views allow interactive exploration of graph clusters.

offer the user the possibility to interactively select a suitable feature set (see the following paragraph) based on a particular use case.

We support interactive selection and weighting of features via the user interface depicted in Figure 3. A set of sliders allows to set weights for each implemented feature. The sum of weights is normalized to 1.0 and changing of a weight of a feature influences the weights of the other features so that the sum stays constant. This allows to create variable user-preferred weighting schemes.

The user interface includes a heatmap visualization of the correlation matrix of the respective features, which helps in selecting features, e.g., the most orthogonal (uncorrelated) ones. The manual feature selection is assisted by simple tools that assess the feature relevance and suggest the user to include or exclude them from the set. The weights of features with zero variance are automatically preset to zero. The features that are highly correlated with many other features are proposed to be excluded from the analysis. More sophisticated semi-automatic feature selections (see [38]) will be included in the future.

All features are *normalized* in order to allow an easy proportional weighting. The normalization is based on graph theoretical aspects. Each feature score is calculated as a fraction of the actual value of the feature, relative to the theoretic maximum value. This yields a $[0,1]$-normalization, which generally gives good results in the SOM clustering according to our observation. Note, another way would be to use the expected value of each feature in random graphs for feature normalization (instead of its maximum theoretic value).

5 Interactive Visualization

In this section, our approach for the interactive visualization of the results of SOM clustering is presented. The results are displayed in a SOM grid and interactive functions allow further exploration of the data space. The assessment of clustering quality is supported by interactive views and calculation of quality metrics.

5.1 Visualization of Clustering Results

The SOM algorithm, equipped with discriminative features, usually provides meaningful results, showing an effective overview of the types of graphs in the data space. *The clustered results* are visualized by showing one representative graph for each cluster on the SOM grid. The representative is chosen as the nearest neighbor sample graph to the respective SOM prototype vector. Figure 4 (center) illustrates a graph cluster map. The background color of each SOM grid cell indicates the relative size of the cluster, measured by the number of sample graphs matched, relative to the maximum number of samples at any SOM grid cell. For large SOM grids, the size of the grid cell can be very small which influences the readability of the displayed graphs. In this case, either the representative graph may be shown on demand in an extra view or hierarchic SOMs can be used.

In order to *explore the members of individual clusters*, the cluster elements are displayed on demand in a so-called *member view*. Figure 4 illustrates member views of several graph clusters, arranged around the cluster map. For clusters with a large number of members (hundreds, thousands), it is possible to visualize the distance distribution or feature distribution of the cluster (see next section) and explore parts of the clusters on demand. Large clusters can also be used as an input for subsequent clustering. This allows for refinement of the clustering results as illustrated in Figure 5.

The visualization of a connected component in the grid employs node-link diagrams with edge width corresponding to the edge weight and arrows showing the edge direction (cf. Figure 4). As the development of a new layout algorithm was not in the focus of the work, standard layouts provided by JUNG [41] were included for visualization of the graphs. The choice of the applied layout is user dependent. It should however support an easy comparison of graphs in the SOM grid. In our case, this works nicely however it cannot be generalized.

The visualization of clustering results is additionally supported

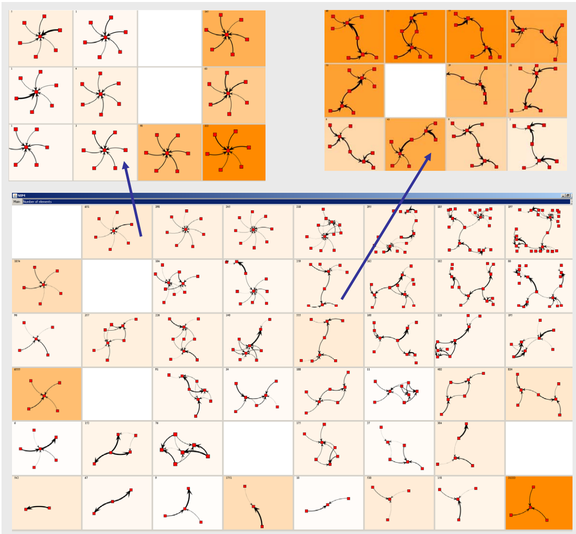


Figure 5: Refinement of SOM. The figures on the top show the SOM grids created by clustering members of the cells in the initial SOM on the bottom of the picture.

by display of so-called *component planes* (cf. Figure 6). This view shows the distribution of the individual features in the resulting SOM matrix. It shows the values of each feature characterizing the cluster center across the SOM grid. The values are displayed as a heatmap. For example, the component plane for the graph size feature (top left), shows that bigger graphs are concentrated in the right up corner of the SOM grid and smaller in the left lower corner. On demand, the values of a selected feature are displayed as the background color of the SOM grid cells.

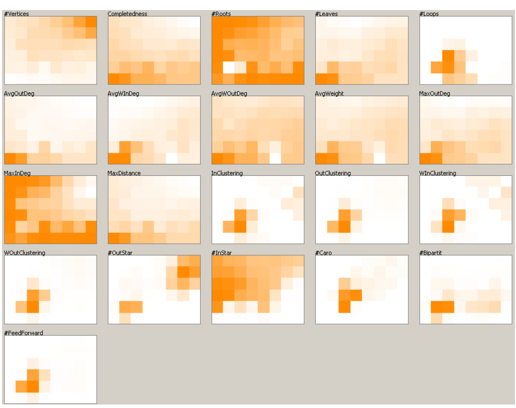


Figure 6: Visualization of component planes.

5.2 Visualization of Clustering Quality

In order to assess the quality of the clustering, several interactive views on the results and calculation and visualization of clustering quality measures are provided. The *distance distribution view* shows the distribution of distances between cluster elements and the respective cluster center (cf. Figure 7). In this view, we can see that the SOM clustering provides very good results, as most of the graphs assigned to a cluster are very close to the cluster center and only few outliers with larger distances appear. The outlier graphs are displayed in a member view, where the background color corresponds to the distance to the cluster center. For comparison, the cluster representative is shown as well (the top left graph).

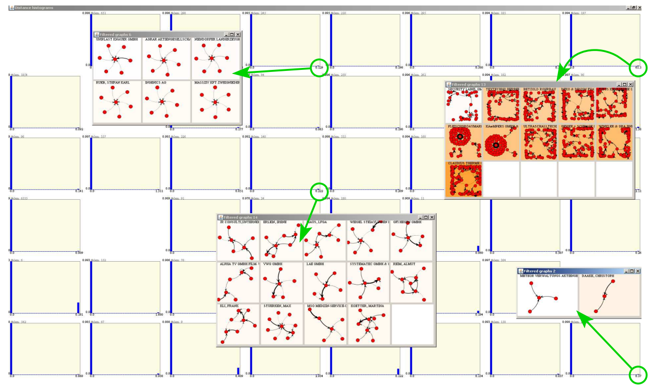


Figure 7: The distribution of distances between cluster members and cluster center are displayed in form of a histogram. The cluster outliers together with cluster representatives are displayed in a separate window on demand, by clicking on the respective histogram bar. Distance to the center is mapped to the background color.

The *display of feature distributions* for the members of the cluster shows the range and frequency of graph features in a selected cluster. Together with the graph distance view (cf. Figure 7), it allows to spot outliers in the clusters and assess the overall quality of the clustering. On demand, an overview of the cluster members across the feature distribution or parts of the histogram can be displayed for detailed analysis of the cluster (cf. Figure 8).

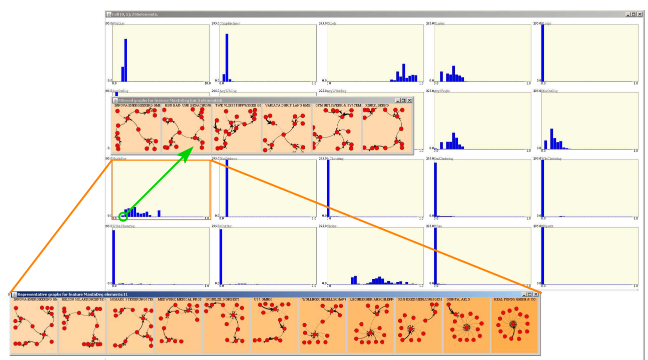


Figure 8: Feature distribution for a particular cluster. An overview of the cluster, showing representant graphs from all parts of the distribution, is displayed in a pop-up window.

In addition to the previous views, the system provides the *assessment of clustering quality by various measures*. The currently used measures follow proposals from various surveys, e.g., [43, 36, 29, 16]. The overview of the measure values together with the distribution of a selected measure across the SOM grid are displayed (cf. Figure 9 for an illustration). This allows for a quantitative assessment of the SOM quality and detailed inspection of the quality measure values across SOM cells.

6 Application

We discuss an application of our system on data of the German corporate shareholding network. Section 6.1 introduces the used data set. Section 6.2 then demonstrates the effect of selecting different feature sets for clustering. Section 6.3 illustrates a corporate structure analysis scenario. Note that, in the following analysis, the chosen SOM grid size is 8 * 6 as it offers reasonable results.

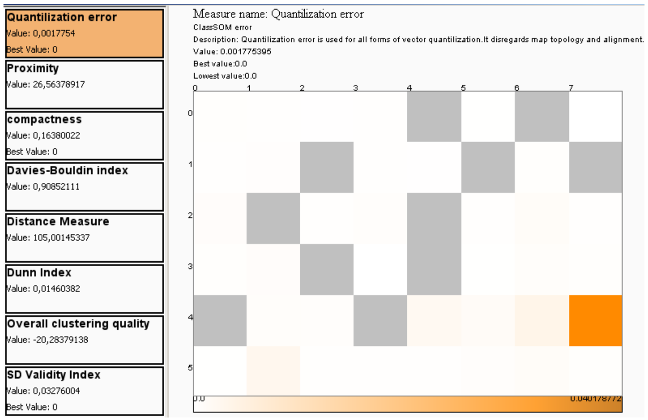


Figure 9: Visualization of selected SOM quality measure.

6.1 Shareholder Structure Data Set

Shareholding relationships between companies in an economic system can be regarded as a directed network with nodes representing companies, and weighted, directed edges representing the "holds-shares-in" relationship between corporations. We consider the *Amadeus* database [10], which contains financial and ownership data on German corporations. The shareholding relationships are provided as tables with lists of companies and private persons holding shares in a company and their amount. The shareholding graph extracted from the database contains more than 300,000 entities. The graph contains one very large connected component with more than 115,000 nodes (135,000 edges), one graph with more than 20,000 nodes and around 40,000 smaller weakly connected components with up to 110 nodes each. For the application, we concentrate only on the part of the data without the two largest components. These components form two separate clusters and can be analyzed separately using specialized graph exploration techniques.

6.2 Interactive Data Analysis

Graph visualization using SOM clustering can be applied to the *analysis of structure types formed in an economic system*. The SOM grid (cf. Figure 4) shows that the shareholding structure sizes in Germany vary from simple 2 node graphs (bottom left corner) to more complex larger graphs (top right corner). The star-shaped graphs are the most important corporate structures, having the highest frequency and occupying multiple cluster centers (with varying graph sizes). The member views allow detailed analysis of individual companies with similar structures.

For the analysis of the data set, we rely on interactive feature selection while using a constant SOM grid size. Changing of feature sets in combination with SOM clustering leads to various views on the data set. These views show the distribution of types of components. It shows which subgraphs are frequent in the data set and which are exceptional under the given feature set. In the following paragraph we describe our findings from the shareholder data set.

Figure 10a shows a SOM produced using only the *number of nodes* as a graph descriptor as a first naive approach. From top-left to bottom-right, graphs of increasing size are arranged on the map. It already reveals what sizes of shareholder structures there are in the dataset. Subgraphs with up to 6 nodes are very frequent. Then, with a gap, larger graphs occur.

Figure 10b shows a map obtained by extending this feature by the *graph completeness*. In effect, the initial coarse SOM layout is refined by accommodating more differentiation regarding a notion of the graphs' complexity. The result shows that the larger the company structures are the more complex is the relationship within them. Small holdings consist mainly of many shareholders of one company and larger include more interwoven cross-holdings.

Finally, Figure 10c shows the usefulness of the feature controlling for *number of loops* in the graph for extracting extraordinary examples of companies holding directly shares in themselves. This phenomena is unexpected. It can be either an outlier, a data quality problem or an interesting company structure which should be reflected in the subsequent detailed analysis.

6.3 Sectoral Analysis of Corporate Structures

The presented approach can be used also for *comparison of corporate structures among several geographic regions* (e.g., USA, Germany, Italy etc.) or for *comparison of industries, types of companies*, etc.. In this paper, we have analyzed the *German industries*. In the dataset, each company is categorized into industries according to the Statistical Classification of Economic Activities in the European Community (NACE). We have used the top level of the classification for the distribution of the company structures into 17 industry categories. In order to compare the types of companies across German industries, we have applied the SOM clustering and visualization to the sets of companies in each industry. The results for the three largest industries using the same set of features are presented in Figure 11. In general, the structural distribution, although having variable layout, is similar to the whole economic system (cf. Fig. 4). Especially, the star-shaped corporate structures appear to be the most frequent in all subsets. However, it can be seen that extended star-shaped structures occur only in manufacturing (cf. Fig. 11a). These star-shaped structured are more complex and seem not to be captured in the other sectors.

7 Conclusions and Future Work

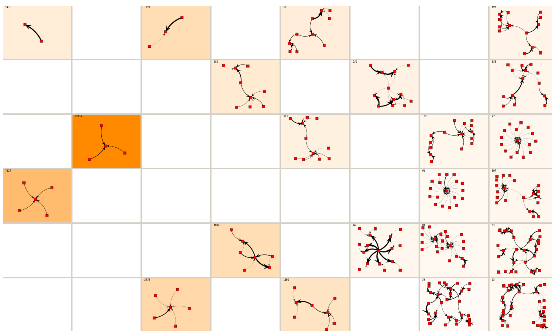
In this paper, we described a novel approach to the visual analysis of graphs with many components. The approach is based on an effective combination of adaptive graph clustering and rich visual-interactive facilities for data exploration. Interactive feature selection for flexible clustering with visual output, and assessment of clustering quality provide comprehensive visual cluster analysis for graph data. The reproducibility and comparability of cluster results is supported by storing analysis parameters and user annotations.

The analysis of large graphs with many weakly connected components is essential in various application areas, e.g., corporate governance, supply chain management, networks of biologic reactions, or cash-flow analysis. We have applied our system on a large data set of German corporate shareholding networks as an example use case.

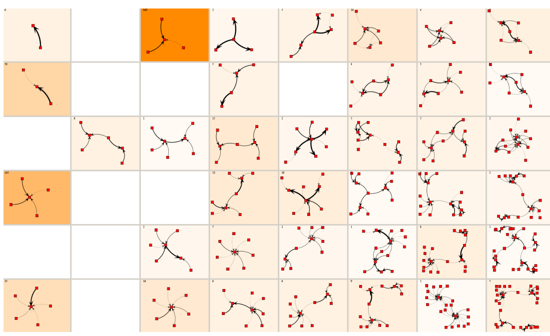
In the future, we will extend the system for supporting comparison of results using also different clustering methods (e.g., k-means, hierarchic clustering). We plan to extend the standard SOM algorithm with a hierarchic version, to provide efficient clustering even in presence of very large data sets and weighting of cluster sizes. We would like to introduce functionality for comparisons of various cluster results and enhance semi-automatic feature selection by using statistic approaches (see [38]). Our system should also be extended by features for other graph types, e.g., labeled graphs (for example biologic reactions). The visualization can be enhanced by a specialized layout supporting comparison of graphs. The sensitivity of our approach w.r.t. various graph data sets, and input parameters will be studied.

Acknowledgements

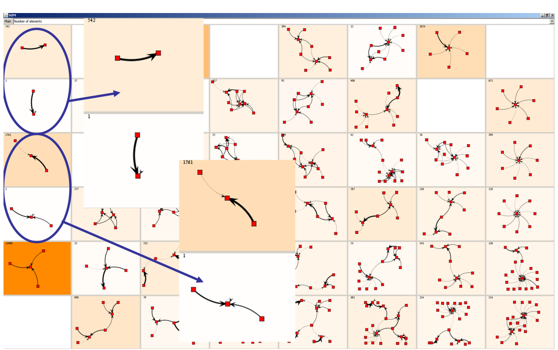
This work was partially supported by the German Research Foundation (DFG) within the project Visual Feature Space Analysis as part of the Priority Program Scalable Visual Analytics (SPP 1335). The authors would like to thank Arjan Kuijper, Torsten Techmann, Martin Hess, Robert Rehner and Martin Ried for their support.



(a) Number of vertices

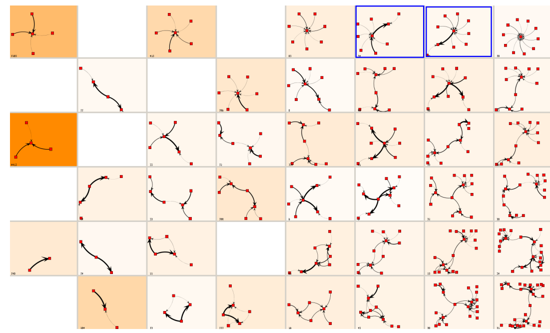


(b) Number of vertices and completeness

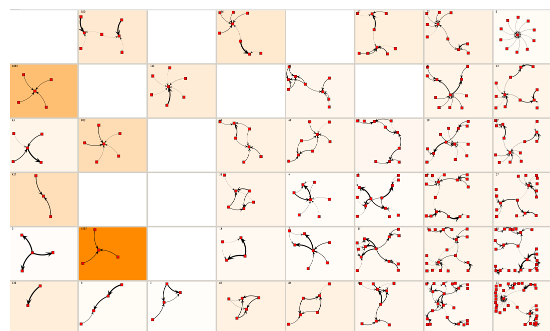


(c) Various selected features including number of loops

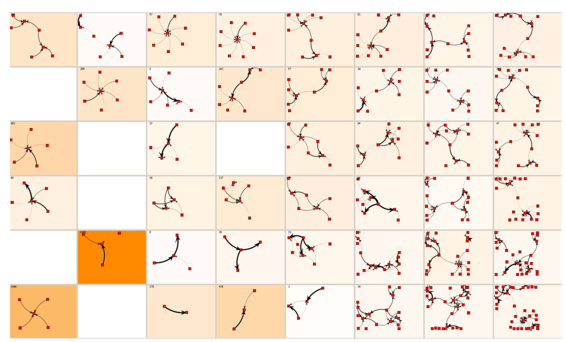
Figure 10: Graph clusterings obtained for various graph feature selections.



(a) Manufacturing (ca. 9,000 components)



(b) Wholesale and retail trade (ca. 10,000 components)



(c) Real estate (ca. 9,000 components)

Figure 11: SOM graph clustering results for corporate structures in German industries.

REFERENCES

[1] A. T. Adai, S. V. Date, S. Wieland, and E. M. Marcotte. Lgl: creating a map of protein function with an algorithm for visualizing very large biological networks. *J Mol Biol*, 340(1):179–190, June 2004.

[2] D. Archambault, T. Munzner, and D. Auber. Topolayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):305–317, 2007.

[3] D. Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale visualization of small world networks. *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 75–81, 2003.

[4] A. Barrat, M. Barthelemy, R. Pastor-Satorras, A. Vespignani, and G. Parisi. The architecture of complex weighted networks. *Proc. of the National Academy of Sciences of USA*, 101(11):3747–3752, 2004.

[5] K. U. Barthel. Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics. *Int. Workshop on Image Analysis for Multimedia Interactive Services*, pages 227–230, 2008.

[6] E. Bonabeau. Graph multidimensional scaling with self-organizing maps. *Int. J. on Information Sciences*, 143(1-4):159–180, 2002.

[7] E. Bonabeau and F. Hhaux. Self-organizing maps for drawing large graphs. *Inf. Proc. Letters*, 67:177–184, 1998.

[8] P. Bonacich. Power and centrality: A family of measures. *The American J. of Sociology*, 92(5):1170–1182, 1987.

[9] U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. volume 2832, pages 568–579. Springer-Verlag, 2003.

[10] Bureau van Dijk Electronic Publishing. AMADEUS Database, Oct. 2007. http://www.bvdep.com/.

[11] G. Caldarelli. *Scale-Free Networks: Complex Webs in Nature and Technology*. Oxford Finance, 4 2007.

[12] A. Capocci, G. Caldarelli, and P. D. L. Rios. Quantitative description and modeling of real networks, 2002.

[13] L. Cohen and T. Deschamps. Grouping connected components using minimal path techniques. application to reconstruction of vessels in 2d and 3d images. volume 2, pages II–102–II–109 vol.2, 2001.

[14] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1277–1284, 2008.

[15] U. Dogrusoz. Two-dimensional packing algorithms for layout of disconnected graphs. *Information Sciences*, 143(1-4):147–158, 2002.

[16] A. Fiannaca, G. Fatta, S. Gaglio, R. Rizzo, and A. Urso. Clustering quality and topology preservation in fast learning soms. In *Proc. 18th international conference on Artificial Neural Networks, Part I*, pages 583–592. Springer-Verlag, 2008.

[17] K. Freivalds, U. Dogrusöz, and P. Kikusts. Disconnected graph layout and the polyomino packing approach. In *Revised Papers from the 9th Int. Symposium on Graph Drawing*, pages 378–391. Springer-Verlag, 2002.

[18] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. of the DIMACS Int. Workshop on Graph Drawing*, pages 388–403, London, UK, 1995. Springer-Verlag.

[19] P. Gajer and S. G. Kobourov. Grip: Graph drawing with intelligent placement. In *GD '00: Proceedings of the 8th International Symposium on Graph Drawing*, pages 222–228, London, UK, 2001. Springer-Verlag.

[20] D. Goehlsdorf, M. Kaufmann, and M. Siebenhaller. Placing connected components of disconnected graphs. In *6th Int. Asia-Pacific Symposium on Visualization 2007*, pages 101–108, Februar 2007.

[21] S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. *Pattern Recogn. Lett.*, 23(4):405–417, 2002.

[22] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kauffman, 2nd edition, 2006.

[23] D. Harel and Y. Koren. Graph drawing by high-dimensional embedding. In *GD '02: Revised Papers from the 10th International Symposium on Graph Drawing*, pages 207–219, London, UK, 2002. Springer-Verlag.

[24] I. Herman, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, /2000.

[25] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[26] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM—self-organizing maps of document collections. In *Proc. Workshop on Self-Organizing Maps (WSOM)*, pages 310–315. Helsinki University of Technology, 1997.

[27] S. H. C. Information and M. Jünge. Drawing large graphs with a potential-field-based multilevel algorithm. *Lecture notes in Computer Science*, 4372:285–295, 2005.

[28] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[29] S. Kaski and K. Lagus. Comparing self-organizing maps. In *ICANN 96: Proceedings of the 1996 International Conference on Artificial Neural Networks*, pages 809–814, London, UK, 1996. Springer-Verlag.

[30] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 1990.

[31] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In *Visual Data Mining*, pages 76–90. 2008.

[32] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

[33] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som_pak: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, 1996.

[34] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812, 2006.

[35] U. Lauther. Multipole-based force approximation revisited - a simple but fast implementation using a dynamized enclosing-circle-enhanced k-d-tree. *Lecture notes in Computer Science*, 4372:20–29, 2007.

[36] C. Legány, S. Juhász, and A. Babos. Cluster validity measurement techniques. In *AIKED'06: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[37] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proc. of the 12th ACM SIGKDD Int. Conference on Knowledge Discovery and data mining*, pages 631–636, New York, NY, USA, 2006. ACM.

[38] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman and Hall, 2007.

[39] B. Meyer. Self-organizing graphs - a neural network perspective of graph layout. In *GD '98: Proc. of the 6th Int. Symposium on Graph Drawing*, pages 246–262, 1998.

[40] M. Neuhaus and H. Bunke. Self-organizing maps for learning the edit costs in graph matching. *Proc. IEEE Trans. on Systems, Man, and Cybernetics*, 35:503–514, June 2005.

[41] J. O'Madadhain, D. Fisher, and S. White. Analysis and visualization of network data using jung. *Journal of Statistical Software*, VV(II).

[42] A. N. Papadopoulos and Y. Manolopoulos. Structure-based similarity search with graph histograms. In *Proc of the 10th Int. Workshop on Database & Expert Systems Applications*, page 174. IEEE Computer Society, 1999.

[43] G. Poelzlbauer. Survey and comparison of quality measures for self-organizing maps. In *WDA 2004*, pages 67 – 82, 2004.

[44] B. Prieto, F. Tricas, J. J. Merelo, A. Mora, and A. Prieto. Visualizing the evolution of a web-based social network. *J. of Network and Computer Apps*, 31:667–698, 2008.

[45] A. R. Rudolf Mayer, Thomas Lidy. The map of mozart. In *Proc. of the 7th Int. Conference on Music Information Retrieval (ISMIR'06)*, pages 351–352, October 8-12 2006.

[46] H. Schwbbermeyer. *Analysis of Biological Networks*, chapter 5, pages 85 – 112. Wiley Series on Bioinformatics, Computational Techniques and Engineering. Wiley, 2008.

[47] J. Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.

[48] K. Zhang, J. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs. *Int. J. of Foundations of Computer Science*, 7(1):43–57, 1996.