Visualization of Small World Networks Using Similarity Matrices

Saima Parveen and Jaya Sreevalsan-Nair

Abstract

Visualization of small world networks is challenging owing to the large size of the data and its property of being "locally dense but globally sparse." Generally networks are represented using graph layouts and images of adjacency matrices, which have shortcomings of occlusion and spatial complexity in its direct form. These shortcomings are usually alleviated using pixel displays, hierarchical representations in the graph layout, and sampling and aggregation in the matrix representation. We propose techniques to enable effective and efficient visualization of small world networks in the similarity space, as opposed to attribute space, using similarity matrix representation. Using the VAT (Visual Assessment of cluster Tendency) algorithm as a seriation algorithm is pivotal to our techniques. We propose the following novel ideas to enable efficient hierarchical graphical representation of large networks: (a) parallelizing VAT on the GPUs, (b) performing multilevel clustering on the matrix form, and (c) visualizing a series of similarity matrices, from the same data set, using parallel sets-like representation. We have shown the effectiveness of proposed techniques using performance measurements of parallel implementation of VAT, results of multilevel clustering, and analyses made in case studies.

I. Introduction

Consider a time series of coauthorship network. Graph layout of such a network can show the "hubs," or nodes with high centrality, e.g., which may indicate faculty members heading labs or research groups. Additionally, it will be insightful to find significant events in the network, e.g., when two authors who worked together earlier have stopped publishing together or two authors who did not work together are collaborating currently. However it will be difficult to infer temporal changes in the network in a single view. Visualizing large networks tend to be a challenge which can be addressed by constructing meaningful hierarchical structures. We propose using similarity-based clustering in the network to identify features such as temporal or spatial events and to construct hierarchical levels of detail.

Similar to [20], we will be using the terms "graph", "vertices", and "edges" to refer to the topological structures with no associated attributes, and "network", "nodes", and "links" to refer to structures related to a graph with attributes, respectively. Elements of an adjacency matrix correspond to vertices, and those of a similarity matrix correspond to nodes.

Our work is based on the premise that similarity functions can be considered to be transformations on the adjacency matrix and can be further used for clustering to manage large data. Several analyses performed on a network, e.g. using similarity functions, centrality measures, etc. can be reduced to being a transformation on the adjacency matrix [26], [6], and hence it is implicitly a transformation on the graph itself. These transformations can be linear or nonlinear. This serves as a motivation for our work, as graphical representation of these transformations can help us gain more insights into the data and our work can extend beyond our current scope of analysis using similarity functions, e.g., analysis based on centrality measures can be made from matrix representations.

For this study, we have focused on small world networks in its undirected form characterized by a local structure where two connected nodes can have several neighbors. Extensive studies on small world networks can be found in [47], [4]. Applying a similarity function, which is symmetric, on an adjacency matrix yields a normalized symmetric matrix called similarity matrix. Though our work can be extended to any network and any normalized symmetric matrix which is characteristic of the network, the performance of our techniques will vary depending on the properties of the network and the transformation applied on the adjacency matrix to obtain the concerned matrix.

We have found that in the information visualization community, matrix visualization has been routinely applied for adjacency matrices for undirected graphs. Adjacency matrices are generally visualized as a binary shaded representation of a matrix, indicating presence or absence of edges between vertices using white and black colors, respectively. Visualizing grayscale matrix representations for similarity matrices has been done before, such as the shaded similarity matrices and variants of the Visual Assessment of cluster Tendency (VAT) algorithm [3], where permuted or seriated similarity matrix representations are used for finding clusters. Using VAT for seriation of the similarity matrix is pivotal to our work. VAT is considered equivalent to a single linkage hierarchical algorithm, which is significant as we use the seriated similarity matrix to construct multiple levels of detail in a hierarchical fashion, for which an underlying single linkage algorithm is apt. Since VAT has a quadratic time complexity, it is inefficient for large scale data sets. Hence we propose a parallel implementation of VAT on the GPUs using CUDA [40].

This document is preprint, as on September-09-2013; Saima Parveen and Jaya Sreevalsan-Nair are with International Institute of Information Technology Bangalore; email: saima.parveen@iiitb.org, jnair@iiitb.ac.in

Multilevel clustering has traditionally been done on the node-link diagrams using nearest neighbor consideration, which is satisfied by VAT owing to its relation to single linkage algorithm. Multilevel clustering performed on the network has been carried out in ways which are specific to research communities: (a) in information visualization community, multilevel clustering of the graph is used for obtaining multiple levels of detail and for further processing pertaining to building focus+context representations, and (b) in machine learning community, one level of clustering is done using similarity functions. We have not found any work pertaining to multilevel clustering in a network combining the two approaches, i.e., obtaining multiple levels of detail using similarity functions. Finding meaningful functions to persist clustering in multiple levels of detail is in itself a challenging problem and is beyond the scope of our current work.

A series of similarity matrices can be generated from a data set in several ways, e.g., every time stamp in time series data, application of different similarity functions on the same data, or different subspace clustering in multivariate data. Finding two or more authors in the same cluster in a coauthorship network over a period of time may indicate that they are working very closely to each other owing to their employment at the same workplace, or having the right set of complementary expertise in pursuing research on the same topic or project. These authors not coauthoring a paper together after a while might indicate that they no longer work together owing to change of employment or change in research interests, etc. Information about such an event becomes very apparent from the clustering tendencies found in similarity matrix series. Tracking membership of objects in clusters across a series of similarity matrices can help in finding trends and patterns, and also in understanding the underlying varying properties of the data. The idea is that if we observe that two objects are present in the same clusters in multiple instances in the series of similarity matrices, we can conclude that the objects are positively correlated with respect to the underlying properties of the different instances. While there is existing research on visualizing similarity matrices for small world networks, we have not found any visual analysis for a series of similarity matrices. This is unique to our work, where we have adapted parallel sets representation to see clustering trends in a series of similarity matrices.

Summarizing, our contributions are as follows:

1) We propose a GPU-based parallel implementation of VAT (pVAT), which is an optimal parallel implementation of VAT using CUDA based on Borůvka's algorithm [44].
2) For data simplification we use the VAT images to perform multilevel clustering, i.e., create multiple levels of detail by recursively clustering the nodes and merging the nodes in the clusters to form new nodes.
3) We propose parallel sets-like representation for tracking the cluster membership of objects in series of similarity matrices generated from the same data, i.e., to view how constituency of the clusters changes across the series, and apply this technique on visualizing small world networks.

II. Related Work

Our design choices have been the result of choosing (a) matrix visualization over graph layouts for small world networks, and (b) VAT algorithm for seriation of similarity matrices. Network visualization and analysis, and that pertaining to small world networks are very active areas of research. Though it will be impossible to refer to the huge body of related literature in this section, we have listed the relevant research ideas that have influenced our work. Though one level of clustering in similarity matrix is routinely used in data mining applications, we did not find any related work on multilevel clustering performed on a similarity matrix. Similarly we have not found any work on analysis of a series of similarity matrices of a multivariate data set, including network data.

Small World Network Visualization:

Watts and Strogatz [47] have explained the characteristics of small world networks based on the small world phenomenon, also known as six degrees of separation [13]. There are several multiscale, multilevel visualizations of graph layout of large-scale small world networks [2], [1], [43], [9], which are scalable applications. These works largely exploit the property of small world networks, i.e. globally sparse and locally dense layout [4], using graph layouts.

Ghoniem et al. [11], [12] have analyzed comparison of graph visualization techniques between node-link diagrams and matrix based representations using controlled experiments. Though the experiments conducted in [11] primarily were for low-level readability tasks and not specifically for social networks, the results are applicable to small world networks. Using experimental results in [11], Ghoniem et al. have shown that node-link diagrams are suited for smaller graphs and graphs that were "almost trees", but performed very poorly for almost complete graphs and denser networks. The idea is that, given the property of small world networks of being globally sparse and locally dense and that matrix visualization is less sensitive to density variations, the matrix visualization will be better suited for small world networks for clarity.

Matrix Visualization of Networks:

Henry and Fekete [18] have proposed a network visualization system, MatrixExplorer, which uses two representations, namely, node-link diagrams and images of adjacency matrices, and have designed the system using inputs from social scientists.Henry and Fekete have used participatory design techniques to arrive at MatrixExplorer. Some relevant results in the form of requirements obtained from the users, who were social science researchers, are: higher preference for matrix-based representations over node-link diagrams as it was faster to display and easier to manipulate for larger data sets;

cluster detection was essential for social networks analysis; and aggregating networks using clusters presented results well. To address the shortcoming of the MatrixExplorer of having huge cognitive load during context switches, Henry and Fekete have further proposed a hybrid representation, which integrates matrix and graph visualizations, called MatLink [19]. Energy-based clustering of graphs with non-uniform degrees (LinLog) algorithm [39] has been used for the node-link representation in MatLink.

Elmqvist et al. [9] have proposed an interactive visualization tool called Zoomable Adjacency Matrix Explorer (ZAME), for exploring large scale graphs. ZAME, which is based on representation of the adjacency matrix, can explore data at several levels of detail. In ZAME, aggregation of nodes is performed based on "symbolic data analysis" and aggregates are arranged into a pyramid hierarchy. Henry et al. have proposed NodeTrix [20] which uses the focus+context technique, where matrix visualization is used for subnetworks while graph layout is used for overview, thus confirming to the locally dense and globally sparse property. ZAME and NodeTrix use nested views [24] without using clustering; whereas our proposed representations use clustering and can be used to obtain juxtaposed views.

While these techniques explore data in the attribute space, our work focuses on exploring the same data in the similarity space. Using relationship-based approach by working in a suitable similarity space, as opposed to the high-dimensional attribute space, has been shown to be more effective for several data mining applications [42]. Additionally, similarity space can be considered to be a superset of attribute space, as identity function on adjacency matrix can be a similarity function.

Seriation Algorithms:

Seriation algorithms are ubiquitous in terms of their applications – Liiv [31] has given an overview of seriation methods, which includes the historical evolution of the technique, various algorithms used for it, and various applications, spanning across several fields, such as, "archaeology and anthropology; cartography, graphics, and information visualization; sociology and sociometry; psychology and psychometry; ecology; biology and bioinformatics; cellular manufacturing; and operations research." Mueller et al. [35], [36] have performed a thorough analysis of vertex reordering algorithms, which is relevant to representation of clusters in matrix visualization. In [35], Mueller et al. have used pixel-level display to scale up the visualization for larger data sets. In [36], Mueller et al. have referred to visual representation of adjacency matrix as visual similarity matrices (VSM) and specified common structural features of the VSM along with their graph-based interpretations. The linked views of the matrix visualization and node-link diagram show that the specific "features" in the VSM correspond to a relational pattern between the vertices in the graph. One such feature is the blocking pattern along diagonal lines, which carry information of clustering tendencies, as showcased in VAT. Though VAT focuses on blocking patterns along the diagonal of the matrix, Mueller et al. have analysed diagonal entities on and off the main diagonals.

VAT:

Bezdek and Hathway [3] have proposed a seriation algorithm along with a visual representation to assess the layout of clusters in the matrix, known as VAT (Visual Assessment of cluster Tendency). VAT uses a grayscale image of a permuted matrix to show blocks along the main diagonal, which are representative of clusters. There have been several improvisations done to VAT owing to its complexity being $O(n^2)$. Successors of VAT relevant to our work are reVAT (revised VAT) [21], bigVAT [22] and sVAT (scalable VAT) [16]. reVAT uses profile graphs to replace the dissimilarity image thus reducing the number of computation. bigVAT and sVAT are scalable variants of VAT based on sampling representative entities to reduce the size of the data. We alternatively propose parallel implementation of VAT (pVAT) where a parallel implementation of Borůvka's algorithm is used for finding minimum spanning trees in the graph [44]. We propose pVAT because sampling algorithms, such as bigVAT and sVAT, rely on good choices of representative data and could lead to inadvertent loss of important data.

Though results of VAT depend on the starting node for the construction of minimum spanning tree, the similarity function used and the inherent clustering in the data, VAT does not involve computation of agglomerative hierarchical data structures, which becomes computationally intensive for large data sets. VAT itself can be considered to be related to single-linkage algorithm [17] as opposed to the average-linkage algorithms used in shaded similarity matrices [45], [46], [48]. Multilevel clustering has traditionally been done on the node-link diagrams using nearest neighbor consideration, which is satisfied by VAT in our proposed algorithm for multilevel clustering owing to relation of VAT to single-linkage algorithm.

Similarity Matrix Visualizations:

Two-dimensional visualization of similarity matrices has been done before, e.g., by Eisen et al. [8]. Wishart has discussed about shaded distance matrices [48] where similarity matrices are reordered by constructing a dendrogram and reordering the dendrogram to minimize the sum of weighted similarities. Wang et al. have used (a) nearest neighbor clustering and ordering using decision tree to visualize a shaded similarity matrix in [46], and (b) a combination of conceptual clustering in machine learning, and cluster visualization in statistics and graphics whose complementary properties help in interpreting clusters better, in [45]. Strehl and Ghosh [41], [42] have proposed Clusion which constructs and reorders a similarity matrix using relationship-based approach for clustering, which is an algorithm called Opposum. Erdelyi and Abonyi [10] have worked on node similarity based visualization which is very similar to ours in the use of VAT for similarity matrix representation.

However they have proposed a new similarity function and implemented dimensionality reduction to obtain a similarity matrix in a lower-dimensional embedding, on which subsequently VAT is used.

These representations use an explicit clustering method which is subsequently used for reordering similarity matrices. Our work focuses on embedding the clustering with the reordering as is done in the VAT algorithm [3].

Parallel Sets:

Inselberg et al. [23] have proposed parallel coordinates for visualizing high dimensional data in a plane. Parallel coordinates representation has been effectively used for several high-dimensional data sets. Inspired by parallel coordinates, Kosara et al. [25] have proposed parallel sets for visualizing and exploring categorical data. While parallel coordinates is for representing points in multi-dimensional or multi-attribute space, parallel sets can be considered for discretely viewing categories. In our work, clusters can be related to categories and hence when visualizing a series of similarity matrices, we propose using a parallel sets-like approach to get an overview of the series.

III. Basics

Similarity matrices and the VAT algorithm, which is a seriation algorithm, are pivotal to our proposed work. We describe the definitions and properties of similarity functions, seriation algorithms, and the VAT algorithm in this section.

A. Similarity Functions

For an ordered set of elements, applying a similarity function on pairwise choice of elements gives an idea of similarities and dissimilarities inherently present in the data. Similarity matrices are two-way one-mode representation of similarity function values between any two elements in the set. "Two-way one-mode" implies that a single set of elements is represented in two different orders, e.g., row and column orders in a matrix [31]. Subsequent clustering on a similarity matrix gives further insight to the data and can be used for reducing the complexity of the data. Similarity matrices are normalized and symmetric.

In graphs, clustering can be done in a similar fashion using similarity matrix whose elements are the vertices of the graph. In most cases the similarity matrix is a function of the adjacency matrix, since both the matrices are two-way one-mode representations of the vertices and functions for certain forms of similarity, such as structural similarity, can transform an adjacency matrix to a similarity matrix [26]. We have used the following standard similarity functions for our experiments: identity, jaccard, dice, inverse log weighted, and cosine similarity. Identity function implies that the similarity matrix is the same as the adjacency matrix.

B. Seriation Algorithms

An algorithm for optimal ordering of entities in a two-way one-mode representation as a square matrix, e.g. the adjacency matrix and similarity matrix, is a permutation algorithm, also known as a seriation algorithm. Ordering a set of N vertices in a graph can be done in $N!$ different ways. Graph theory states that ordering of vertices to optimize a cost function is called minimum linear arrangement (MINLA/MinLA), which is a known NP-hard problem. Hence, one uses heuristic algorithms to solve ordering of vertices to achieve domain-specific optimization criteria. An ideal ordering algorithm should be linear or better, in terms of runtime complexity. The choice of the starting vertex is critical to most ordering algorithms. We assume node 0 as starting vertex and have used the following permutation algorithms for our experiments: VAT, reVAT, Breadth first search (BFS), Depth first search (DFS), Reverse Cuthill-Mckee (RCM) [5], Kings [34], and Modified minimum degree (MMD) [32].

Figure 1 shows the effect of applying the permutation algorithms on the subnetwork of the condensed matter coauthorship network in 1999 [37]. For sake of clarity in showing cluster tendencies, we have chosen a subnetwork of 372 nodes from the coauthorship network which has 16725 nodes. We observe that VAT helps in identifying well-defined clusters, better than the other algorithms. As shown in Figure 1, the blue highlight shows a natural cluster in VAT, which the other seriation algorithms fail to identify clearly.

C. VAT

VAT [3] uses Prim's algorithm for finding the minimum spanning tree (MST) of a weighted graph to permute the order of elements in the similarity matrix. VAT shows clear clusters as black blocks along the diagonal when the matrix is represented using a grayscale image. However VAT works only on symmetric matrices and also, fails for cases where (a) there are no natural clusters in the data, and (b) the cluster types can not be identified using single linkage.

Why VAT ?:

VAT has several advantages which are apt for our application on small world networks. VAT neatly arranges clusters along the main diagonal, which helps in visual assessment as well as isolating the clusters. VAT is considered to be a single-linkage algorithm owing to its dependence on a minimum spanning tree. This aligns with the multilevel clustering where we implicitly build a hierarchical structure.

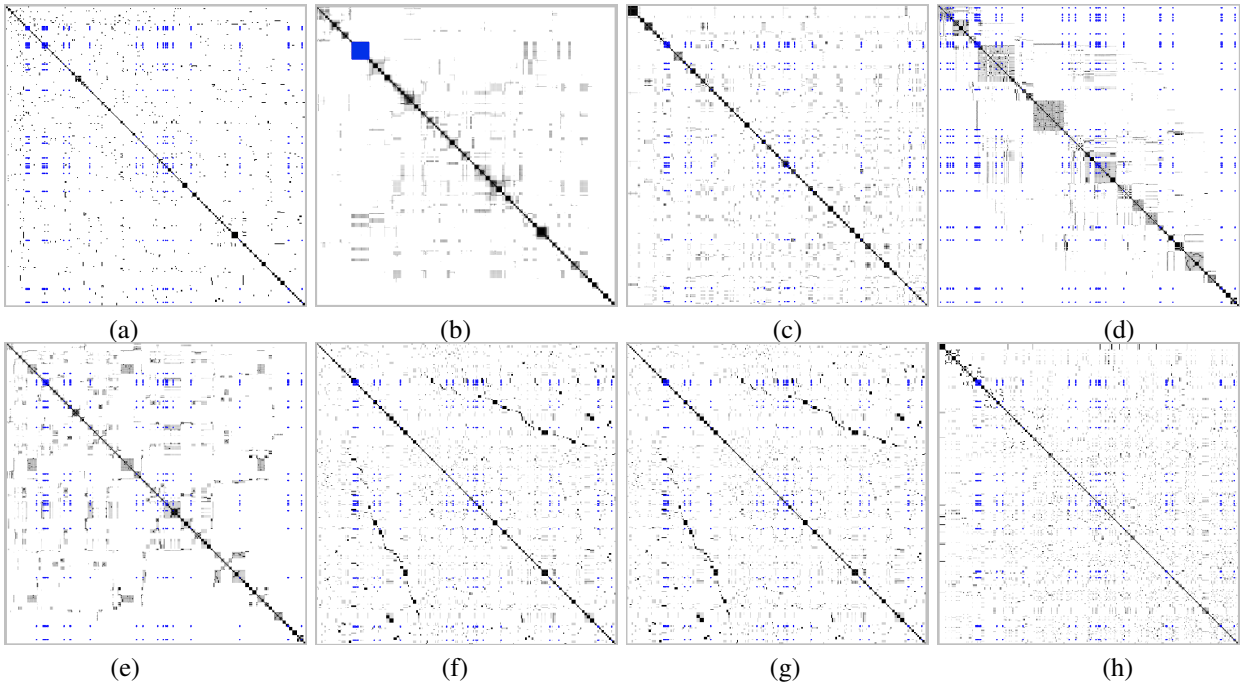

Fig. 1. Permutation or seriation algorithms showing clustering tendencies in a similarity matrix generated using cosine similarity function for a subnetwork of 372 nodes in the condensed matter collaboration network in 1999 [37]: (a) shows the unseriated similarity matrix, and (b)-(g) show the similarity matrix seriated using VAT, reVAT, BFS, DFS, RCM, Kings, and MMD algorithms, respectively. The blue highlight shows a natural cluster, which is identified by VAT but not the other seriation algorithms.

IV. Parallel Implementation of VAT

Since VAT has a runtime complexity of $O(N^2)$ for N nodes, it is not scalable for large data sets. Though bigVAT [22] and sVAT [16] are scalable, they are sampling algorithms which rely on good choices of representative data. Inappropriate choices of samples in these algorithms can result in wrong results. Hence we propose the parallel implementation of VAT (pVAT).

The serial (original) implementation of VAT uses Prim's algorithm for constructing the MST for finding the permuted order of elements in the similarity matrix. For the parallel implementation, we propose using Borůvka's algorithm for finding the MST as implemented by Vineet et al. [44] on the GPU. Borůvka's approach is generally favored in parallel implementation owing to its iterative nature. The runtime complexity of Borůvka's algorithm is $O(E \log V)$ for a graph with V nodes and E edges, while that of Prim's algorithm is $O(V^2)$, which implies that implementation of pVAT depends on both E and V, while that of original VAT depends only on V.

In VAT, we exclusively order the vertices without constructing the actual MST or evaluating the feasibility of constructing one. In a disconnected graph, a MST cannot be constructed but we can still get an ordering and construct the ordered image in VAT, which is an advantage. Both Prim's and Borůvka's algorithms give us combinatorial possibilities of MSTs. Our proposed algorithm for pVAT is as given in Algorithm 1.

A. Locating Clusters

The original implementation of VAT does not automatically differentiate the clusters. However for our algorithms for multilevel clustering as well as for parallel sets-like representation we require specific clusters. We use a brute-force approach as follows: in the reordered dissimilarity matrix D^* obtained using VAT, we walk along the diagonal rowwise, check columnwise the intensities of the neighboring elements to the diagonal element in the next row in D^*, and use a heuristically-derived threshold to identify start and end of clusters along the diagonal. The runtime complexity of the algorithm is $O(N^2)$ for $N \times N$ dissimilarity matrix.

V. Multilevel Clustering

In our work, we use multilevel clustering using an agglomerative hierarchical approach for achieving multiple levels of detail. It works in a bottom-up manner, initialized by all data objects as leaf nodes. For implementing multilevel clustering, we merge all the nodes in a cluster into a single node, when moving from a finer to a coarser level of detail, and iteratively apply clustering algorithm on the new set of nodes. The multilevel clustering algorithm terminates when none of the nodes can be further merged, i.e, there are no more clusters. For N nodes $(x_1, x_2, \ldots, x_N)$, if we get k clusters $(c_1, c_2, \ldots, c_k)$ after applying similarity function and seriation algorithm, where $k < N$, then the k clusters become the nodes for the subsequent

Algorithm 1: pVAT: Parallel implementation of VAT

Input : D – $N \times N$ dissimilarity matrix
Output: Reordered Dissimilarity D^*
compute weighted graph G(V,E) with N vertices in V and edges in E, obtained from using D as the adjacency matrix.
$P = \{\}$
$S = V$
for *each vertex* u *in* V **do**
  for *each vertex* v *in* V*, such that* $v \neq u$ **do**
    find the minimum weighted edge from u to v
while *no more vertices* $u \in S$ *can be merged* **do**
  merge vertices ($u \in U \subseteq S$) to form connected components, called supervertices (sv_U), using minimum weighted edges
  treat supervertices as new vertices, $S = S \cup \{sv_U\} - U$
for *each vertex* u *in* S **do**
  get the recursive ordering $O(u)$ of the subgraph in u
  $P \leftarrow P \cup O(u)$
obtain the ordered dissimilarity matrix D^* using the ordering array P as: $D^*_{pq} = D_{P(p),P(q)}$ for $1 \leq p, q \leq N$.

level of detail. In general, multilevel clustering is possible only if $k < N$ strictly, as $k = N$ would imply that every cluster is a singleton, which implicitly indicates that there are no inherent clusters in the data set.

When moving to a coarser level of detail, the merged nodes replace the constituent nodes. For sake of simplicity, we will assume a merged node to contain one or more nodes. Thus all the nodes in the subsequent coarser level of detail are merged nodes. When moving to a coarser level of detail, the adjacency matrix changes, and the similarity matrix needs to be recomputed. An appropriate formulation for the attributes of a merged node has to be derived from an appropriate aggregation of the attributes of the constituent nodes from the finer level, which is beyond the scope of this work. Hence in our current work, a simple similarity function of finding the maximum weighted edge between the clusters has been used. It is equivalent to the identity similarity function, i.e., the adjacency and similarity matrices are the same. This function reduces a $n \times n$ matrix to a $k \times k$ matrix, where n nodes have reduced to k clusters across one level of detail.

When using VAT as the seriation algorithm, any one of the possible minimum spanning trees is used which makes the results of our multilevel clustering combinatorial. The number of clusters for a particular combination of data set, similarity function and seriation algorithm may vary for all transitions to the coarser levels except the last one. We have observed that the last transition always gives the same result, as the multilevel clustering terminates when no further merges can occur, i.e., the last level of detail occurs when the data set cannot be further simplified.

Multilevel clustering accentuates the need for node-cluster labelling as the new aggregated nodes should be meaningful in the context of the data set. Our work, however, is currently limited to tracking the membership of a node in a hierarchical data structure generated by multilevel clustering. The rationale behind this decision is that the membership of nodes in clusters, on its own right, implicitly reveals analytical information.

VI. Visualization of Similarity Matrix Series

Similarity matrix series refers to multiple similarity matrices obtained for a set of data objects, either in the form of time series, or by application of various similarity functions or permutation functions, or by generating different subspace clusterings. Membership of objects in clusters in each of these similarity matrices is a salient aspect of the series as well as of the entire data set. Tracking the cluster-membership of objects will enable us in identifying trends and patterns in the data, thus showing the evolution of the data object across the series. We propose a parallel sets-like representation for tracking cluster-membership of objects across similarity matrices. We believe representing small world networks using this technique can give us further insights.

Kosara et al. have proposed parallel sets representation [25] of categorical data. Parallel sets has the following features: (a) it shares the parallel coordinate layout, treating all dimensions to be independent of each other, and (b) instead of using a line connecting values of coordinates to represent a data point, boxes are used to display categories and parallelograms or bands between the axes to show the relations between categories.

In our representation, the axes indicate the "instances" in the series, e.g. time-stamps in time series, one of the similarity functions in the series obtained by applying various functions, etc. The axes show the permuted order of objects in the seriated similarity matrix of the corresponding instance. We use segments in the axes to indicate clusters and lines between axes to link locations of an object in the permuted order in the different instances, similar to boxes and parallelograms in the parallel sets representation. Hence we refer to our technique as "parallel sets-like." For a series of similarity matrices, each similarity

matrix can be considered to be in its own independent space, which justifies using linearly independent axes to represent them. As shown in Figure 2, each of the parallel axes corresponds to a $N \times N$ similarity matrix in the series and displays the ordered set of N data objects or nodes. Each node in the matrix has a label which can be tracked. Segments in the axes are assigned colors to indicate start and end of clusters. We use a two-color scheme where the colors are assigned alternately along each of the parallel axes to indicate start and end of clusters.

Generating such series of similarity matrices for a small world network and representing the series using our proposed method can aid in data mining.

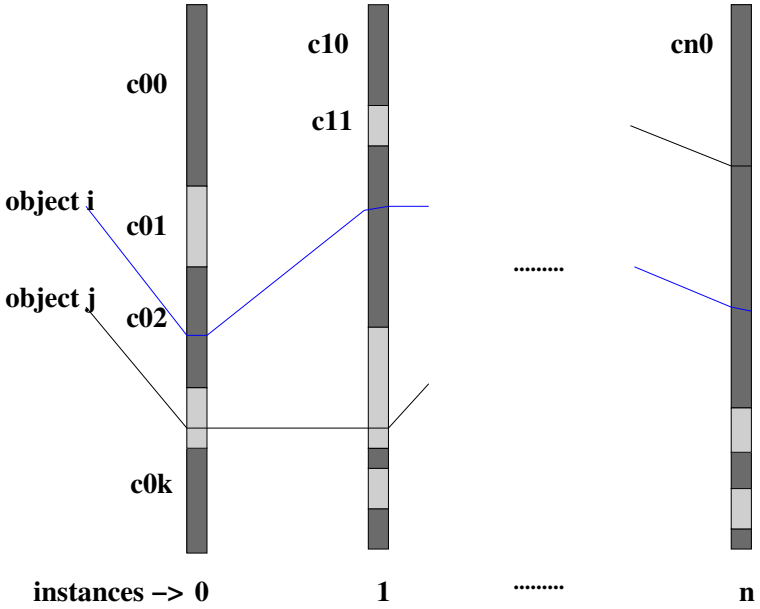

Fig. 2. Schematic of parallel sets-like representation of similarity matrices series. The parallel axes indicate the different instances $(0, 1, \ldots, n)$, which correspond to the similarity matrices in the series. The similarity matrices represented here have been permuted using a seriation algorithm, e.g. VAT, and clusters have been identified in each matrix. For each instance a, the axis is segmented using a two-color scheme to show start and end of clusters; c_{ab} indicates the b^{th} cluster in the seriated similarity matrix a. The lines representing objects start from an initial order in the left of the axes and connect points on the axes corresponding to the object in the permuted order on the specific axis, as shown by the blue and black lines indicating objects i and j, respectively.

VII. Experiments & Results

In this section, we have described the experiments we conducted to show the effectiveness of our techniques and have shown the results for the same. We have performed case studies to show the insights we drew from using our techniques.

pVAT:

For evaluating pVAT, we have used the following data sets: coauthorship network [33], snapshots of peer-to-peer file sharing network [27], [28], who-votes-on-whom network data [30], and cocitation network [29]. Implementation of pVAT algorithm requires CUDA [40] and CUDPP [15] libraries. We have also used the igraph library [7] for graph layouts and implementing similarity functions. We have performed the experiments using Nvidia GeForce GTX 480 and 4GB RAM.

Performance measurements comparing pVAT to the serial implementation of VAT on different data sets, after applying jaccard similarity function, are as given in Table I, which shows as expected that the former is more efficient than the latter. It is to be noted that serial implementation of VAT involves the Prim's algorithm whereas pVAT is based on Borůvka's algorithm. Hence, we see a difference in the trends in our experiments and the speedups from the implementation in Vineet et al. [44], where serial and parallel implementations of Borůvka's algorithm are compared. Since the algorithms used for serial and parallel implementations are different, it will be incorrect to give speedups, as would be the case in a classical parallel computing scenario. Owing to the difference in dependency of the algorithms in the serial and parallel implementations on V and E, for a graph of V vertices and E edges, we observe the following from Table I:

1) MST construction for a graph of 7K vertices and 103K edges will be faster than 8K vertices and 26K edges for the serial version, as there is no dependency of the runtime complexity on the number of edges.
2) MST construction for a graph of 7K vertices and 103K edges will be slower than a graph of 10K vertices and 52K edges for the parallel version, as there is a linear dependency on number of edges but only a logarithmic dependency on the number of vertices.

Multilevel clustering:

Data set	#Nodes	#Links	Serial VAT (msec)	pVAT (msec)
Coauthorship network [33]	475	1250	$1x10^3$	2.42
Peer-to-peer file sharing network [27]	6301	20777	$1.8x10^6$	4.55
Who-votes-on-whom network [30]	7115	103689	$2.4x10^6$	5.21
Peer-to-peer file sharing network [28]	8114	26013	$3.8x10^6$	4.07
Cocitation network [29]	9877	51971	$6.6x10^6$	3.92

TABLE I
PERFORMANCE MEASUREMENTS OF SERIAL AND PARALLEL IMPLEMENTATIONS OF VAT FOR VARIOUS NETWORK DATA SETS, AFTER APPLYING THE JACCARD SIMILARITY FUNCTION.

Data set	#Nodes	#Nodes in Level Transitions	#Levels
Karate club[49]	22	22 $\rightarrow$10$\rightarrow$ 5$\rightarrow$4	3
Taro exchange[14]	34	34 $\rightarrow$7$\rightarrow$2	2
Coauthorship[38]	1589	1589 $\rightarrow$ 615 $\rightarrow$238 $\rightarrow$122 $\rightarrow$113	5
Wikipedia voting[30]	7115	7115 $\rightarrow$3216 $\rightarrow$1983 $\rightarrow$927 $\rightarrow$ 725 $\rightarrow$ 635	6

TABLE II
SIMPLIFICATION OF NETWORK DATA SETS USING MULTILEVEL CLUSTERING BASED ON VAT ALGORITHM FOR SERIATION AND CLUSTER IDENTIFICATION AFTER ITERATIVELY APPLYING IDENTITY FUNCTION ON THE ADJACENCY MATRIX.

For implementing multilevel clustering, we have used the following data sets: coauthorship network [38], who-votes-on-whom network data [30], taro exchange network [14], and karate club network [49]. We have used the identity function on the adjacency matrix as the similarity function and VAT as the seriation algorithm. At every transition in the level of detail, we have performed the following steps: (a) compute the new similarity matrix, (b) merge nodes in cluster to form new nodes, and (c) apply VAT on the new similarity matrix. Table II summarizes the number of nodes in transitions between levels of detail when implementing multilevel clustering.

Case Study 1: Time-series of condensed matter collaboration network [37]:

In order to isolate stories of relevance in large networks, we have chosen the condensed matter collaboration networks [37] in 1999, 2003, and 2005, and created time series of similarity matrices. We have analyzed a subnetwork in condensed matter collaboration network in 1999, 2003, and 2005, which was seeded from subnetworks involving 27 authors who were common in the three data sets. We have isolated 1036 authors in total from all three instances, and applied our analysis based on parallel sets-like representation, as shown in Figure 3, where pink, blue, and red lines track authors in clusters found in 1999, 2003, and 2005, respectively.

The cluster found in 1999 shown in pink in Figure 3 comprises of the following authors: LaSota, Krakauer, Yu, and Cohen. Krakauer advised LaSota and Yu for their Ph.D. and postdoctoral research respectively circa 1999. Krakauer worked on NRL (Naval Research Lab)-funded project and Cohen worked at NRL, circa 1999. These strong proximities led to several coauthored papers amongst these four authors. In 2003, LaSota, Krakauer, and Yu continue to be in a cluster, and Cohen falls off, indicating that the connection through NRL has faded. In 2005, LaSota, Yu, and Cohen are not coauthoring with Krakauer any more, however they still form a cluster, owing to the fact that though they continue to be in the academia in the area of physics, they are no longer active in research. The insights we derived based on coauthorship can be made directly from the graph layout, however, our techniques have additionally enabled us to gain similarity-based insights on the temporal behavioral patterns of the subjects.

The cluster in 2003 highlighted in blue consists of the following authors: Kes, Zeldov, Rappaport, Myasoedov, Feldman, Beek, Avraham, Khaykovich, Shritkman, Tamegai, and Li; who are associated with Zeldov's superconductivity lab in various capacities since 2000. Since the lab is fairly young, the associations are strong in 2003 as well as in 2005, but are scattered in 1999. The strong associations in 2003 could also have derived directly from the network owing to a paper coauthored by all of them in 2002; however the associations in 2005 is not obvious from the network directly.

The cluster in 2005 highlighted in red consists of the following authors: Wang, Junod, Bouquet, Toulemonde, Weber, Eisterer, Sheikin, Plackowski, and Revaz. They are in the same cluster in 2003, indicating that all the authors are associated through Junod by coauthorship from 2003. This inference made with the help of the parallel sets-like representation has been validated by the actual data.

Since the clusters are formed based on the cosine similarity function, we find that given a "hub", such as Zeldov and Junod, all their coauthors tend to cluster together even in the years the coauthors themselves do not publish together. This is because cosine similarity measure is based on the common neighbors and the hubs continue to act as common neighbors.

Case Study 2: Analysis of a Collaboration Network [33]:

We have applied both parallel sets-like representation and multilevel clustering on a collaboration network of social network analysts [33]. We have generated a series of similarity matrices by applying the following similarity functions: jaccard, dice,

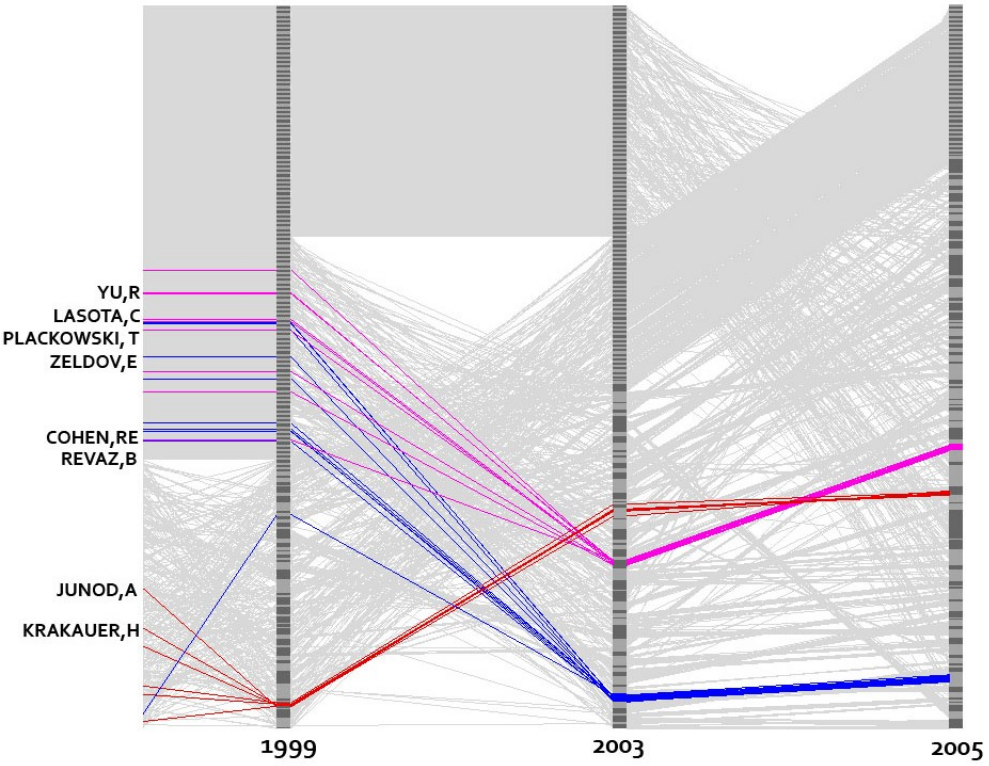

Fig. 3. Parallel sets-like representation of a subnetwork of 1036 authors taken from the condensed matter collaboration network in 1999, 2003, and 2005, to generate time series of similarity matrices. We have analyzed the clusters isolated in 1999, 2003, and 2005, highlighted by pink, blue, and red, respectively.

cosine similarity, and inverse log-weighted. Parallel sets-like representation of this series, as shown in Figure 4 has shown us the influence of the various similarity functions on the data set. The red highlight shows a cluster that persists through all the similarity functions. On further investigation, we find that the cluster comprises of the following authors: Darrow, Muth, Vera, Marrow, Rothenberg, Daniels, Mann, Liebow, Myers, Branch, Mueller, and Potterat; who have co-authored with a researcher, Klovdahl, who is not in the cluster. Since the four similarity functions are based on common neighbors, the authors' association with Klovdahl causes the cluster to occur on application of each of the similarity functions.

On applying multilevel clustering on iterative application of identity similarity function and VAT, 475 nodes reduced to 217, 203, and 201, in subsequent levels of detail, as shown in Figure 5.

VIII. Conclusions

There have been several improvements to VAT since its inception to handle scalability. However since its scalable variants, namely bigVAT [22] and sVAT [16], are sampling algorithms, we have proposed a parallel implementation of VAT, pVAT, using CUDA and Borůvka's algorithm. pVAT enables preserving all data. As expected, performance of pVAT is more efficient compared to the serial implementation VAT which makes our tool scalable when using VAT as a seriation algorithm. The original VAT algorithm is not intended for network data, hence, no analysis of the algorithm has been done for various kinds of networks. Given the "globally sparse, locally dense" nature of small world networks, and runtime complexity of Borůvka's algorithm being $O(E \log V)$, for a graph of V vertices and E edges, we can conclude that VAT based on Borůvka's algorithm is scalable for our specific application of small world networks.

We have proposed using multilevel clustering on the similarity matrix as a way of reducing large data sets. We have found that irrespective of the variations in the results obtained from VAT, the network attains the same coarsest level of detail. Multilevel clustering, upon applying appropriate aggregation function, will enable meaningful grouping of data to achieve various levels of detail.

We have proposed using a parallel sets-like representation to visually explore multidimensional data which can be posed as a series of similarity matrices. This representation enables us to track the membership of data objects in clusters, that are identified across different similarity matrices, which correspond for different time stamps, dimensions or attributes, characteristic function, such as similarity function or seriation algorithm. Parallel sets-like representation has helped us to find significant events in time series data and interesting behavior in series derived from applying various similarity functions.

Since our work heavily depends on VAT, it comes with its limitations as well. Our future work will involve adapting our methods to address the following shortcomings: (a) representation of dense graphs as well as networks without any inherent

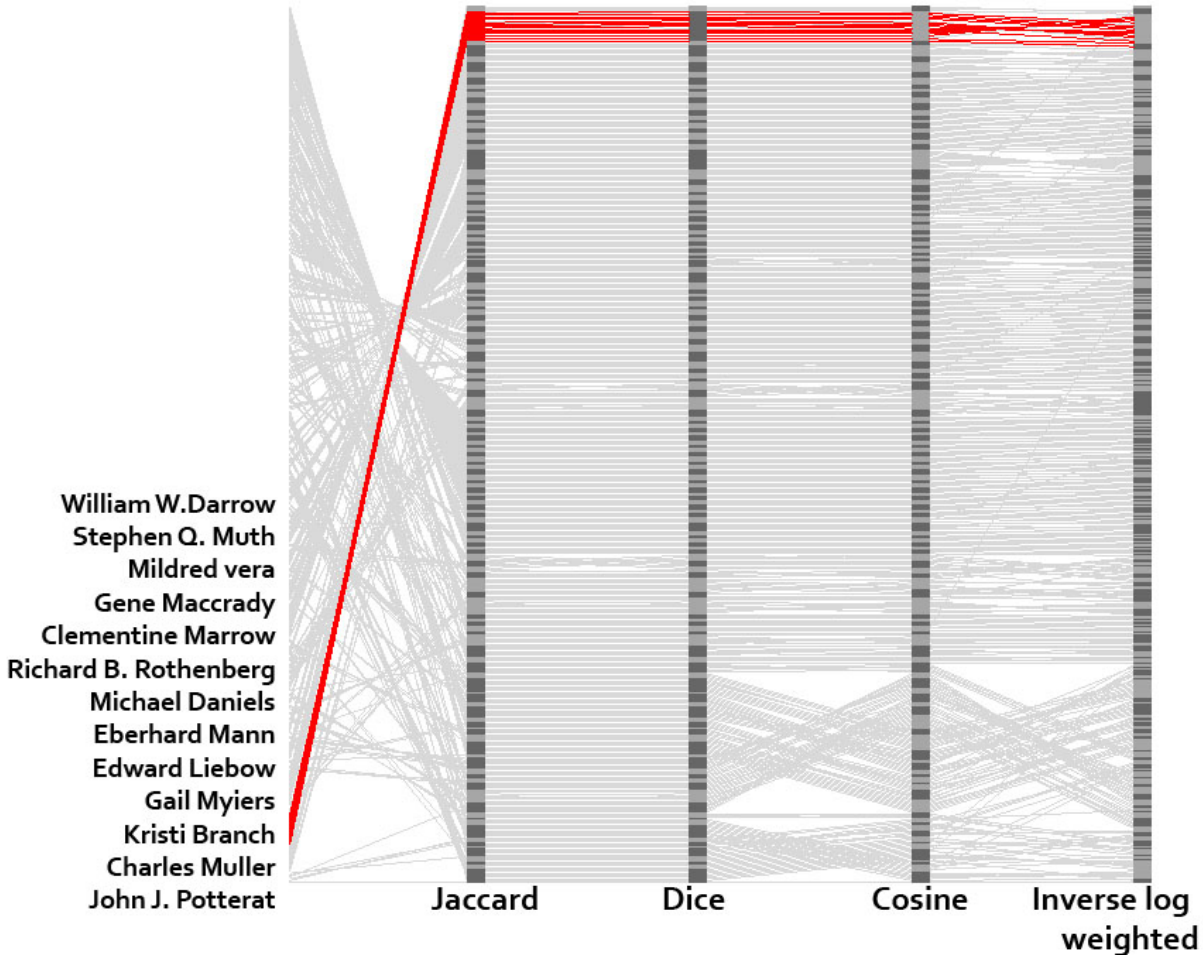

Fig. 4. Parallel sets-like representation of a series of similarity matrices generated from applying various similarity functions and VAT in the collaboration network of social network analysts [33]. The red highlight shows the cluster that persisted through the different similarity function.

clusters, and (b) theoretical or empirical evaluation of our assessment of the clustering capability of the similarity functions. For highly dense networks with large number of nodes where node-link diagram suffers from occlusion, matrix visualization is apt. However highly dense graphs will inherently have fewer number of clusters which will cause multilevel clustering to fail. Hence we will have to adapt our methods to work effectively on dense networks, e.g. VAT may have to be substituted by a suitable permutation algorithm for such graphs. Though matrix representation is limited by a spatial complexity of $O(N^2)$ and is not scalable with data size, it can be resolved using pixel-level displays. Our current work does not rigorously check if VAT gives the correct number of clusters as can be obtained when applying a similarity function on the data. In order to allow for such an evaluation, we will have to formulate appropriate metrics based on our method.

ACKNOWLEDGEMENTS

The authors would like to thank Srujana Merugu, Yedendra Shrinivasan, and Vijay Natarajan for giving valuable feedback on this work. The authors would like to thank IIIT-Bangalore for their support.

REFERENCES

[1] David Auber. *Tulip : A huge graph visualisation framework*. P. Mutzel and M. Junger, 2003.
[2] David Auber, Yves Chiricota, Fabien Jourdan, and Guy Melançon. Multiscale Visualization of Small World Networks. In *Proceedings of the Ninth annual IEEE conference on Information visualization*, INFOVIS'03, pages 75–81, Washington, DC, USA, 2003. IEEE Computer Society.
[3] J. C. Bezdek and R. J. Hathaway. VAT: A Tool for Visual Assessment of (Cluster) Tendency. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2225–2230, Piscataway, NJ, 2002. IEEE Press.
[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, February 2006.
[5] W.M. Chan and Alan George. A linear time implementation of the reverse cuthill-mckee algorithm. *BIT Numerical Mathematics*, 20(1):8–14, 1980.
[6] Carlos D. Correa, Tarik Crnovrsanin, and Kwan-Liu Ma. Visual Reasoning about Social Networks Using Centrality Sensitivity. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):106–120, 2012.

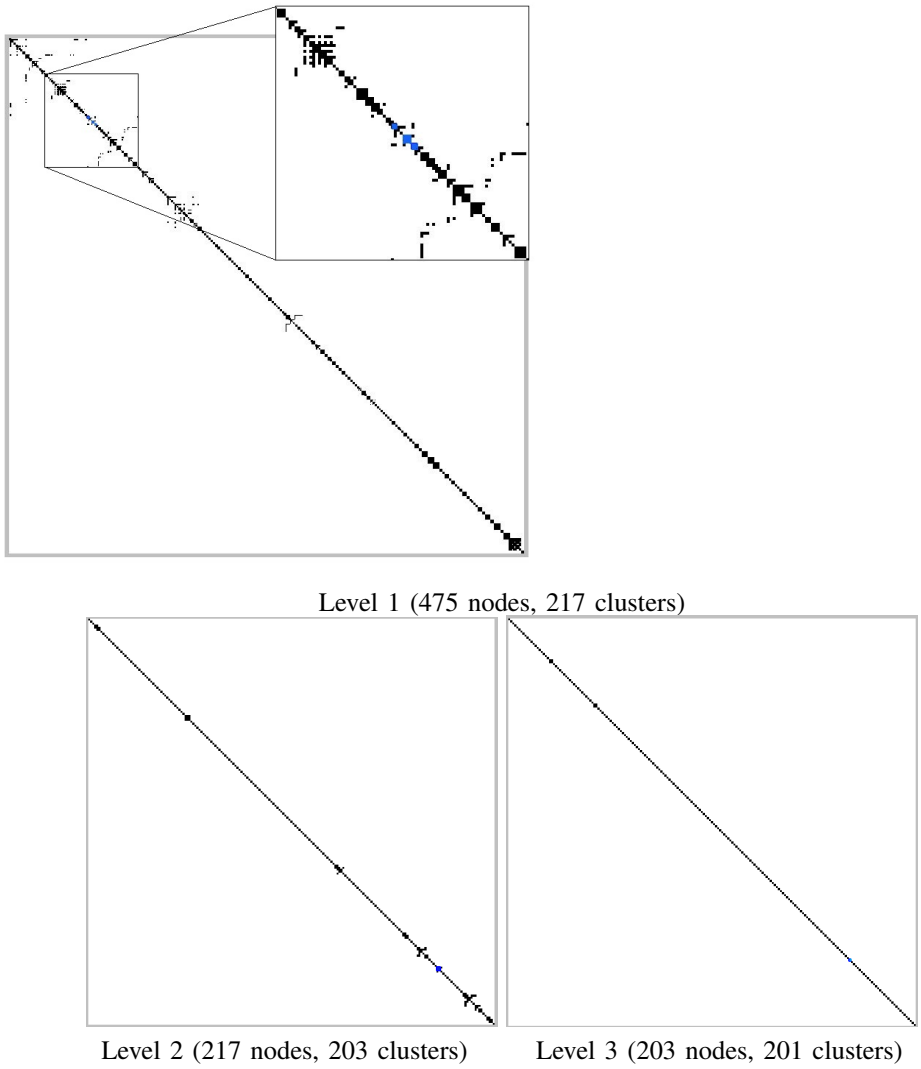

Fig. 5. Multilevel clustering of the collaboration network of social network analysts [33] using identity similarity function and VAT seriation algorithm. The blue highlight shows the nodes that merged to form clusters, which become nodes in the subsequent coarser level of detail.

[7] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[8] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster Analysis and Display of Genome-wide Expression Patterns. In *Proceedings of the National Academy of Sciences of the USA*, volume 95, pages 14863–14868, 1998.

[9] Niklas Elmqvist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and Jean-Daniel Fekete. ZAME: Interactive Large-Scale Graph Visualization. In IEEE Press, editor, *IEEE Pacific Visualization Symposium 2008*, pages 215–222, Kyoto, Japan, 2008. IEEE.

[10] Miklós Erdélyi and János Abonyi. *Node Similarity-based Graph Clustering and Visualization*, page 483494. Citeseer, 2006.

[11] M. Ghoniem, J.-D. Fekete, and P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24, 0-0 2004.

[12] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. On the Readability of Graphs using Node-link and Matrix-based Representations: A Controlled Experiment and Statistical Analysis. *Information Visualization*, 4(2):114–135, 2005.

[13] John Guare. *Six Degrees of Separation: A Play*. Vintage Books, 1990.

[14] P. Hage and F. Harary. Data set of Taro exchange, 1983.

[15] Mark Harris, Shubho Sengupta, John Owens, Yao Zhang, Andrew Davidson, and Nadathur Satish. CUDPP, 2007.

[16] Richard J. Hathaway, James C. Bezdek, and Jacalyn M. Huband. Scalable Visual Assessment of Cluster Tendency for Large Data Sets. *Pattern Recogn.*, 39(7):1315–1324, July 2006.

[17] Timothy C. Havens, James C. Bezdek, James M. Keller, Mihail Popescu, and Jacalyn M. Huband. Is vat really single linkage in disguise? *Annals of Mathematics and Artificial Intelligence*, 55(3-4):237–251, April 2009.

[18] Nathalie Henry and Jean-Daniel Fekete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Trans. Vis. Comput. Graph.*, 12(5):677–684, 2006.

[19] Nathalie Henry and Jean-Daniel Fekete. MatLink: Enhanced Matrix Visualization for Analyzing Social Networks. In *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, INTERACT'07, pages 288–302, Berlin, Heidelberg, 2007. Springer-Verlag.

[20] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, November 2007.

[21] J. M. Huband, J.C. Bezdek, and R.J. Hathaway. Revised Visual Assessment of (Cluster) Tendency (reVAT). In *North American Fuzzy Information Processing Society (NAFIPS)*, pages 101–104, Banff, Canada, 2004. IEEE Press.

[22] Jacalyn M. Huband, James C. Bezdek, and Richard J. Hathaway. bigVAT: Visual Assessment of Cluster Tendency for Large Data Sets. *Pattern Recogn.*, 38(11):1875–1886, November 2005.

[23] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization '90*, VIS '90, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[24] Waqas Javed and Niklas Elmqvist. Exploring the Design Space of Composite Visualization. In *Proceedings of the IEEE Pacific Symposium on Visualization*, pages 1–8, 2012.

[25] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, July 2006.

[26] EA Leicht, Petter Holme, and MEJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

[27] Jure Leskovec. Gnutella peer-to-peer file sharing network, August 08, 2002, 2002.

[28] Jure Leskovec. Gnutella peer-to-peer file sharing network, August 09, 2002, 2002.

[29] Jure Leskovec. High-energy physics citation network, January 1993-April 2003, 2003.

[30] Jure Leskovec. Wikipedia vote network, 2008.

[31] Innar Liiv. Seriation and Matrix Reordering Methods: An Historical Overview. *Stat. Anal. Data Min.*, 3(2):70–91, April 2010.

[32] Joseph W. H. Liu. Modification of the minimum-degree algorithm by multiple elimination. *ACM Trans. Math. Softw.*, 11(2):141–153, June 1985.

[33] Chris McCarty. Data set of network of coauthorships in the Social Networks journal in 2008, 2008.

[34] Chris Mueller. Sparse matrix reordering algorithms for cluster identification. 2004.

[35] Christopher Mueller, Benjamin Martin, and Andrew Lumsdaine. A Comparison of Vertex Ordering Algorithms for Large Graph Visualization. In *Asia-Pacific Symposium on Visualization*, pages 141–148, February 2007.

[36] Christopher Mueller, Benjamin Martin, and Andrew Lumsdaine. Interpreting Large Visual Similarity Matrices. In *Asia-Pacific Symposium on Visualization*, pages 149–152, February 2007.

[37] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, January 2001.

[38] M. E. J. Newman. Data set of coauthorship network in network theory and science, 2006.

[39] Andreas Noack. Energy-based Clustering of Graphs with Nonuniform Degrees. In *Proceedings of the 13th international conference on Graph Drawing*, GD'05, pages 309–320, Berlin, Heidelberg, 2006. Springer-Verlag.

[40] Nvidia. CUDA, 2013.

[41] Alexander Strehl and Joydeep Ghosh. Relationship-based visualization of high-dimensional data clusters. In *Proc. Workshop on Visual Data Mining (KDD 2001), San Francisco*, pages 90–99. ACM, August 2001.

[42] Alexander Strehl and Joydeep Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS J. on Computing*, 15(2):208–230, April 2003.

[43] Frank van Ham and Jarke J. van Wijk. Interactive Visualization of Small World Graphs. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 199–206, Washington, DC, USA, 2004. IEEE Computer Society.

[44] Vibhav Vineet, Pawan Harish, Suryakant Patidar, and P. J. Narayanan. Fast Minimum Spanning Tree for Large Graphs on the GPU. In *Proceedings of the Conference on High Performance Graphics 2009*, HPG '09, pages 167–171, New York, NY, USA, 2009. ACM.

[45] Jun Wang, Bei Yu, and Les Gasser. Classification Visualization with Shaded Similarity Matrix. Technical report, GSLIS, University of Illinois at Urbana-Champaign, 2002. 9 pages.

[46] Jun Wang, Bei Yu, and Les Gasser. Concept Tree Based Clustering Visualization with Shaded Similarity Matrices. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 697–701, Washington, DC, USA, 2002. IEEE Computer Society.

[47] Duncan J. Watts and Steven H. Strogatz. Collective Dynamics of 'Small-world' Networks. *Nature*, 393(6684):440–442, June 1998.

[48] David Wishart. ClustanGraphics3: Interactive graphics for cluster analysis. *Classification in the Information Age, Springer-Verlag*, pages 268–275, 1999.

[49] W. Zachary. Data set of a university karate club, 1977.