PIWI: Visually Exploring Graphs Based on Their Community Structure

Jing Yang, Yujie Liu, Xin Zhang, Xiaoru Yuan, *Member, IEEE*, Ye Zhao, *Member, IEEE*, Scott Barlowe, and Shixia Liu, *Member, IEEE*

Abstract—Community structure is an important characteristic of many real networks, which shows high concentrations of edges within special groups of vertices and low concentrations between these groups. Community related graph analysis, such as discovering relationships among communities, identifying attribute-structure relationships, and selecting a large number of vertices with desired structural features and attributes, are common tasks in knowledge discovery in such networks. The clutter and the lack of interactivity often hinder efforts to apply traditional graph visualization techniques in these tasks. In this paper, we propose PIWI, a novel graph visual analytics approach to these tasks. Instead of using Node-Link Diagrams (NLDs), PIWI provides coordinated, uncluttered visualizations, and novel interactions based on graph community structure. The novel features, applicability, and limitations of this new technique have been discussed in detail. A set of case studies and preliminary user studies have been conducted with real graphs containing thousands of vertices, which provide supportive evidence about the usefulness of PIWI in community related tasks.

Index Terms—Information visualization, visual analytics, graph visualization, community structure

1 Introduction

Biological, social, technological, and information networks can be studied as graphs, and graph analysis has become crucial in understanding the features of these systems [14]. Visual analytics is an important approach to graph analysis, since it allows human beings to combine their domain knowledge and perception abilities with computational power in the reasoning process. Although a large number of graph visualization systems have been developed, many real-world exploratory missions are still in dire need of effective visual analytics tools. A significant one is community related graph visual analysis tasks.

In many real networks, the distribution of edges among the vertices displays big inhomogeneities. There are high concentrations of edges within special groups of vertices, and low concentrations between these groups. This feature of real networks is called community structure [17]. These groups are called communities, clusters, or modules. More generally, communities are groups of vertices which probably share common properties and/or play similar roles within a graph [14]. The community structure has brought significant advances to the understanding of complex systems and has concrete applications [14]. However, the support from existing graph visualization systems for community related tasks is fairly limited. Such tasks include but are not limited to:

Community browsing: Communities and the relations among them provide a coarse-grained depiction of a graph. However, browsing communities is not an easy task for most existing graph visualization systems. For example, to get a general idea of vertices in the communities, users usually need to zoom into the communities one by one to read vertex labels in an often cluttered display.

Community relationship exploration: It is important to examine the relationships among communities to learn how these substructures interact with each other. For example, public crisis managers examine the effectiveness of communications among social communities during disasters. Funding agencies evaluate interdisciplinary research among multiple research areas. In these tasks, the users desire not only statistical metrics summarizing the relationships, but also structural details such as how and through which vertices the communities are connected. Few existing graph visualization systems allow users to flexibly explore such community-reliance relationships with functions that enable a realistic knowledge discovery process.

Attribute-structure relationship analysis: In real networks, vertices usually carry multidimensional attributes or additional complex information. Analyzing the relationship between these attributes and communities is a critical yet difficult task. For example, the correlation between online relationships (e.g., friendship links on Flickr) and offline attributes (e.g., the camera brand owned by a user) is important for personalized advertising in social networks [31]. Users suffer from a lack of effective visual reasoning approaches for exploring such information in enough detail.

Scalable interactions: Users often need to identify and manipulate a large number of vertices with desired

- *J. Yang and Y. Liu are with the Department of Computer Science, College of Computing and Informatics, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223-0001. E-mail: {Jing.Yang, yliu39}@uncc.edu.*
- *X. Zhang and X. Yuan are with the Key Laboratory of Machine Perception (Minister of Education), Center for Information Science, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R. China. E-mail: {zhang.xin, xiaoru.yuan}@pku.edu.cn.*
- *Y. Zhao is with the Department of Computer Science, Kent State University, Kent, OH 44242-0001. E-mail: zhao@cs.kent.edu.*
- *S. Barlowe is with the Department of Computer Science, College of Computing and Informatics, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223-0001. E-mail: sabarlow@uncc.edu.*
- *S. Liu is with Microsoft Research Asia, No. 5 Danling Street, Haidian District, Beijing 100080, P.R. China. E-mail: shixia.liu@microsoft.com.*

Manuscript received 13 Sept. 2011; revised 20 Mar. 2012; accepted 25 July 2012; published online 15 Aug. 2012.
Recommended for acceptance by J.-D. Fekete.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-09-0225.
Digital Object Identifier no. 10.1109/TVCG.2012.172.

attributes and structural features. For example, identifying vertices lying at the boundaries between communities allows analyzers to learn which vertices are playing an important role of mediation and leading the relationships and exchanges between different communities [14]. Bulk interactions, which allow users to identify and manipulate multiple vertices and edges according to given attribute and structural criteria simultaneously, are a direly needed yet less studied feature in graph visualization systems.

In this paper, we propose a novel visual analytics tool named **PIWI**. It enables users to conduct the aforementioned community related tasks effectively. It employs uncluttered, intuitive, and yet meaningful visual representations and interactions to support effective and efficient visual exploration of graphs with thousands of vertices. Its design goals include the following essential features missing in most existing graph visualization systems:

Uncluttered display: Conducting the aforementioned tasks requires simultaneous examination of lots of information, such as the labels and attributes of all vertices connecting multiple communities. Such detailed information, when displayed on a large number of vertices, will clutter most graph visualization tools. A clutter-free examination can greatly increase the applicability of a graph visualization system.

Acknowledging context: Conducting the aforementioned tasks also requires examining details within context. For example, users may want to examine how vertices with desired attributes are distributed among communities. Although it is not difficult to provide contextual information for one or two vertices, most existing visualization systems do not allow users to examine a large number of vertices within context simultaneously. It is therefore important to provide contextual information for a large number of foci simultaneously to enable effective comparison and other visual analytics tasks.

Flexible, effective, and efficient interactions: Interactive graph exploration often requires selections and other manipulations on multiple vertices concurrently. Criteria for such selections are often a combination of structural features and vertex attributes. Although a few approaches have been proposed [24], [36], [3], their functionality and flexibility are limited. New interaction techniques for complex selection criteria are desired.

Supporting effective visual sense making: Sense making often requires users to perform a sequence of investigative tasks in a step-by-step manner. It is also an iterative process in which insights and hypotheses are gradually formed, retrospected, compared, and refined with human input. Graph visualization systems should allow users to use the output of performed steps as the input in further examination. They should also enable users to compare, refine, and integrate intermediate results from multiple visual exploration steps.

2 Related Work

2.1 Node-Link Diagrams

Most existing graph visualization techniques are Node-Link Diagrams (NLDs) or Adjacency Matrix Representations (AMRs). NLDs usually place graph vertices on a 2D or 3D layout area, and explicitly draw the edges among them as links on the display. NLD visualization systems, such as GraphVis [13] and NodeXL [3], can efficiently draw a graph with thousands of vertices in a few seconds. However, overcoming severe graph clutter and providing effective interactive exploration remain significant challenges [24]. To address these challenges, efforts such as visualizations based on the community structure of graphs, progressive visual exploration, and novel interaction techniques have been made.

Community detection has been employed in NLDs to reduce clutter and enable the discovery of community related insights. Vizster [19] visually highlights communities in an NLD and allows users to interactively examine clustering results at different granularities. SocialAction [26] allows users to aggregate networks based on their community structure and examine communities of interest in detail. Itoh et al. [22] hierarchically cluster multiple-category graphs based on the categories and connectivity of vertices. They use a hybrid space filling and force-directed layout to visually reveal communities and their relationships. NodeXL [3] provides an NLD layout style where communities are displayed within boxes so that the connections between the communities can be visually explored. However, none of the existing approaches provide enough visualization and interaction support to community related analysis tasks.

The progressive visual exploration strategy has been practiced by several recent graph visualization systems. Jigsaw [32] allows users to identify vertices of interest from a set of coordinated views and progressively examine their neighborhoods in an NLD. Van Ham and Perer [35] propose to "search, show context, expand on demand" when exploring large graphs. Their approach allows users to browse the immediate context graph around a specified vertex of interest. PIWI supports progressive visual exploration by allowing users to interactively create new vertex groups, which can be visually explored together with automatically detected communities for further analysis.

A handful of interaction techniques have been integrated into NLDs. McGuffin and Jurisica [24] allow users to select a vertex's neighborhood by dragging out its radius using a radial menu besides the traditional rectangle and lasso selection. NodeXL [3] allows users to select multiple vertices using a rectangle. It also supports union and difference of several selections. Viau et al. [36] propose FlowVizMenu and Parallel Scatterplot Matrix, where vertex metrics are displayed and interactive selections based on these metrics are conducted. PIWI supports intersection, along with union and difference, of arbitrary groups (both those automatically generated by the system and those created on the fly by users). It integrates selections upon individual vertices, attributes, communities, user-defined vertex groups, and neighborhoods of vertex groups. To the best of our knowledge, PIWI is among the first graph visualization systems that provide such rich selection capability.

2.2 Adjacency Matrix Representations

AMRs are visual representations of the adjacency matrices of graphs. Bertin first introduces visual matrices to represent networks [10]. Matrix Zoom [7] displays large graphs as a hierarchy of adjacency matrices given a predefined cluster hierarchy using edge aggregation. MatrixExplorer [20] coordinates AMR and NLD views and supports matrix reordering, interactive filtering, and clustering. NodeTrix [21] is a hybrid visualization with NLDs showing the global structure of a network and AMRs showing communities. Ghoniem et al. [16] show that AMRs outperform NLDs for large graphs or dense graphs in several low-level reading tasks, but not in path-related tasks. PIWI and AMRs both avoid using links in order to reduce clutter. However, the visual metaphor and interactions of PIWI are significantly different from those of AMRs.

2.3 Multivariate Graph Visualization

Many efforts have been made to visualize graphs with multidimensional vertex attributes. PivotGraph [37] aggregates the graph based on vertex attributes and uses a grid-based approach to exploring the relationship between vertex attributes and connections. OntoVis [28] uses the ontology associated with a semantic network to conduct structural abstraction and importance filtering. Pretorius and van Wijk [27] use interactive attribute-based clustering for visualizing large state transition graphs. Semantic substrates [30], [8] position vertices in nonoverlapping regions based on their attributes. They also provide dynamic query sliders and filtering buttons for interactive vertex selection and filtering. GraphDice [11] uses multidimensional visualizations to facilitate multivariate graph visualization. PIWI supports multivariate graph visualization with a distinct new approach: the display of the graph structure is integrated with that of vertex attributes, where users can seamlessly explore them in a unified way.

2.4 Other Clustering-Based Visualizations

Clustering analysis has been widely used in visual analytics for data types besides graphs, such as image collections, document collections, and multidimensional data sets. Previous work reveals that for many real-world visual analytics tasks, especially those high-level tasks such as browsing a large number of data items, visualizations, and interactions at the cluster level should have a higher priority than those on the inner structure of the clusters. For example, Hierarchical Parallel Coordinates [15] visually presents clusters of multidimensional data items using a band and a mean without displaying individual data items in them. Newdle [39] clusters news articles based on an article graph constructed according to tag co-occurrence, represents the clusters using word clouds of their most significant tags, and allows users to interactively explore the structural relationships among the clusters. Scatter/Gather [18] allows users to browse a document collection in topic-coherent groups where the titles, rather than the inner relationships among the documents in the groups, are visualized. The community level visualization and interactions in PIWI are inspired by the above approaches.

In addition, clustering-based document visualizations, such as Newdle [39] and Scatter/Gather [18], reveal the importance of semantic representation in text intensive data sets. Graphs with meaningful vertex and edge labels are text intensive data. PIWI significantly improves the readability of vertex labels and thus make the semantics conveyed by a large graph visible to users.

3 PIWI IN A NUTSHELL

This section presents the visualizations and interactions of PIWI, whose usefulness is exemplified by the visual exploration of a real network. This network, named the NYT graph, is a tag co-occurrence network with 1,200 vertices (tags) and 7,042 edges (co-occurrence of two tags in at least one article). It conveys the tag co-occurrence information of 1,078 New York Times (NYT) [5] world news articles published from February 22, 2011 to April 25, 2011. It also has 36 binary vertex attributes carrying categorical and temporal information. Its communities reflect the major news events reported in the news corpus. *The details of the examples can be seen from the supplementary video,* which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.172.

3.1 Community Detection and Visual Representation of Communities

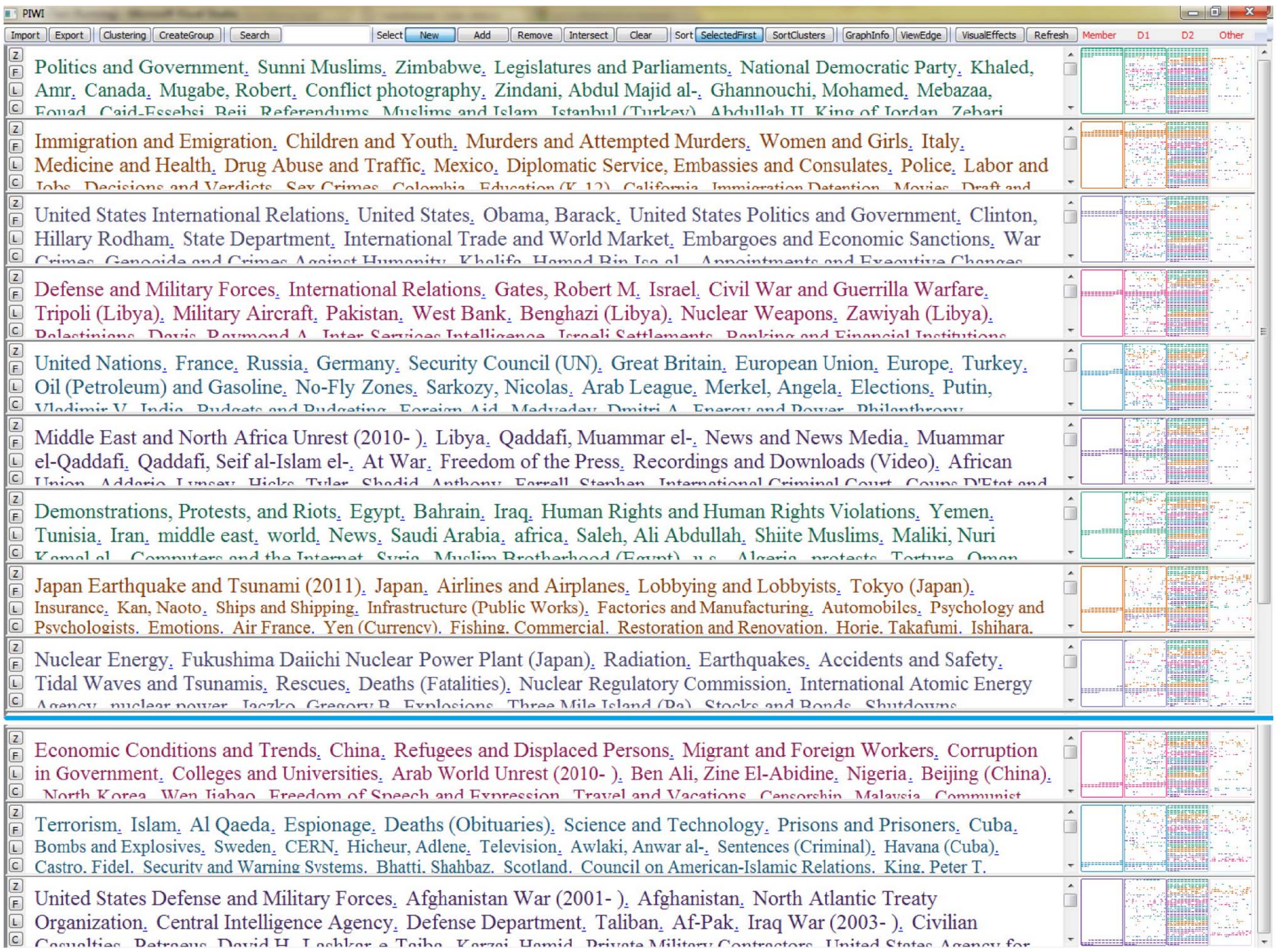
Following the suggestion by Fortunato [14], PIWI visually depicts communities and their relationships to provide a coarse-grained overview for a large graph. The communities can be detected using one of the community detection algorithms integrated into PIWI or imported from external programs. PIWI integrates multiple algorithms, including Girvan-Newman [17] and Clauset-Newman-Moore [12] from the SNAP library [4] and MCL [34]. PIWI allows overlapping communities [25].

In the initial view of a graph, the communities are visually presented to users. Fig. 1 shows the initial view of the NYT graph. Twelve communities are detected using an external graph partition algorithm named AdjCluster [38]. The three communities under the blue line are accessed using the scrollbar on the right of the window; this removes the limitations on how many communities can be displayed and how much space can be assigned to each community.

Each community is assigned a color and represented by a row consisting of a tag cloud and a set of vertex plots. A vertex belonging to the community is represented by its label in the tag cloud and dots in the vertex plots. The label and dots are displayed in the color assigned to the community unless the vertex belongs to multiple communities. In the latter case, the color of a vertex will be the color assigned to the community it is most related to, which is determined by the number of member vertices it connects to or other measures.

In Fig. 1, several distinct colors are assigned to adjacent communities. The colors are reused when there are more communities. They help users distinguish adjacent communities in the vertex plots. Users can interactively change the number of distinct colors or display all labels in gray through the visual effect setting dialog, which is triggered by the "Visual Effects" button on the toolbar.

Tag clouds. A tag cloud is a list of tags where the sizes of the tags reflect their popularity [9]. The tags can be sorted alphabetically or by size. In PIWI, a tag cloud is used to display the labels of all member vertices of a community; the vertices in the central position of the community are emphasized. They may have an important function of control and stability within the community and are often looked for in real applications [14]. In particular, the tag sizes are decided by a user selected centrality metric. For



Politics and Government. Sunni Muslims. Zimbabwe. Legislatures and Parliaments. National Democratic Party. Khaled, Amr. Canada. Mugabe, Robert. Conflict photography. Zindani, Abdul Majid al-. Ghannouchi, Mohamed. Mebazaa, Fouad. Caid-Essebsi, Beji. Referendums. Muslims and Islam. Istanbul (Turkey). Abdullah II, King of Jordan. Zebari

Immigration and Emigration. Children and Youth. Murders and Attempted Murders. Women and Girls. Italy. Medicine and Health. Drug Abuse and Traffic. Mexico. Diplomatic Service, Embassies and Consulates. Police. Labor and Jobs. Decisions and Verdicts. Sex Crimes. Colombia. Education (K-12). California. Immigration Detention. Movies. Draft and

United States International Relations. United States. Obama, Barack. United States Politics and Government. Clinton, Hillary Rodham. State Department. International Trade and World Market. Embargoes and Economic Sanctions. War Crimes, Genocide and Crimes Against Humanity. Khalifa, Hamad Bin Isa al-. Appointments and Executive Changes

Defense and Military Forces. International Relations. Gates, Robert M. Israel. Civil War and Guerrilla Warfare. Tripoli (Libya). Military Aircraft. Pakistan. West Bank. Benghazi (Libya). Nuclear Weapons. Zawiyah (Libya). Palestinians. Davis, Raymond A. Inter-Services Intelligence. Israeli Settlements. Banking and Financial Institutions

United Nations. France. Russia. Germany. Security Council (UN). Great Britain. European Union. Europe. Turkey. Oil (Petroleum) and Gasoline. No-Fly Zones. Sarkozy, Nicolas. Arab League. Merkel, Angela. Elections. Putin, Vladimir V. India. Budgets and Budgeting. Foreign Aid. Medvedev, Dmitri A. Energy and Power. Philanthropy

Middle East and North Africa Unrest (2010-). Libya. Qaddafi, Muammar el-. News and News Media. Muammar el-Qaddafi. Qaddafi, Seif al-Islam el-. At War. Freedom of the Press. Recordings and Downloads (Video). African Union. Addario, Lynsey. Hicks, Tyler. Shadid, Anthony. Farrell, Stephen. International Criminal Court. Coups D'Etat and

Demonstrations, Protests, and Riots. Egypt. Bahrain. Iraq. Human Rights and Human Rights Violations. Yemen. Tunisia. Iran. middle east. world. News. Saudi Arabia. africa. Saleh, Ali Abdullah. Shiite Muslims. Maliki, Nuri Kamal al-. Computers and the Internet. Syria. Muslim Brotherhood (Egypt). u.s.. Algeria. protests. Torture. Oman

Japan Earthquake and Tsunami (2011). Japan. Airlines and Airplanes. Lobbying and Lobbyists. Tokyo (Japan). Insurance. Kan, Naoto. Ships and Shipping. Infrastructure (Public Works). Factories and Manufacturing. Automobiles. Psychology and Psychologists. Emotions. Air France. Yen (Currency). Fishing, Commercial. Restoration and Renovation. Horie, Takafumi. Ishihara.

Nuclear Energy. Fukushima Daiichi Nuclear Power Plant (Japan). Radiation. Earthquakes. Accidents and Safety. Tidal Waves and Tsunamis. Rescues. Deaths (Fatalities). Nuclear Regulatory Commission. International Atomic Energy Agency. nuclear power. Jaczko, Gregory B. Explosions. Three Mile Island (Pa). Stocks and Bonds. Shutdowns

Economic Conditions and Trends. China. Refugees and Displaced Persons. Migrant and Foreign Workers. Corruption in Government. Colleges and Universities. Arab World Unrest (2010-). Ben Ali, Zine El-Abidine. Nigeria. Beijing (China). North Korea. Wen Jiabao. Freedom of Speech and Expression. Travel and Vacations. Censorship. Malaysia. Communist

Terrorism. Islam. Al Qaeda. Espionage. Deaths (Obituaries). Science and Technology. Prisons and Prisoners. Cuba. Bombs and Explosives. Sweden. CERN. Hicheur, Adlene. Television. Awlaki, Anwar al-. Sentences (Criminal). Havana (Cuba). Castro, Fidel. Security and Warning Systems. Bhatti, Shahbaz. Scotland. Council on American-Islamic Relations. King, Peter T.

United States Defense and Military Forces. Afghanistan War (2001-). Afghanistan. North Atlantic Treaty Organization. Central Intelligence Agency. Defense Department. Taliban. Af-Pak. Iraq War (2003-). Civilian Casualties. Petraeus, David H. Lashkar-e-Taiba. Karzai, Hamid. Private Military Contractors. United States Agency for

Fig. 1. The initial view of the NYT graph (1,200 vertices, 7,042 edges). Twelve communities are displayed, each one represented by a row. The tag cloud shows the vertex labels of the community and the vertex plots show its members and neighbors. The communities under the blue line are accessed using the scrolling bar.

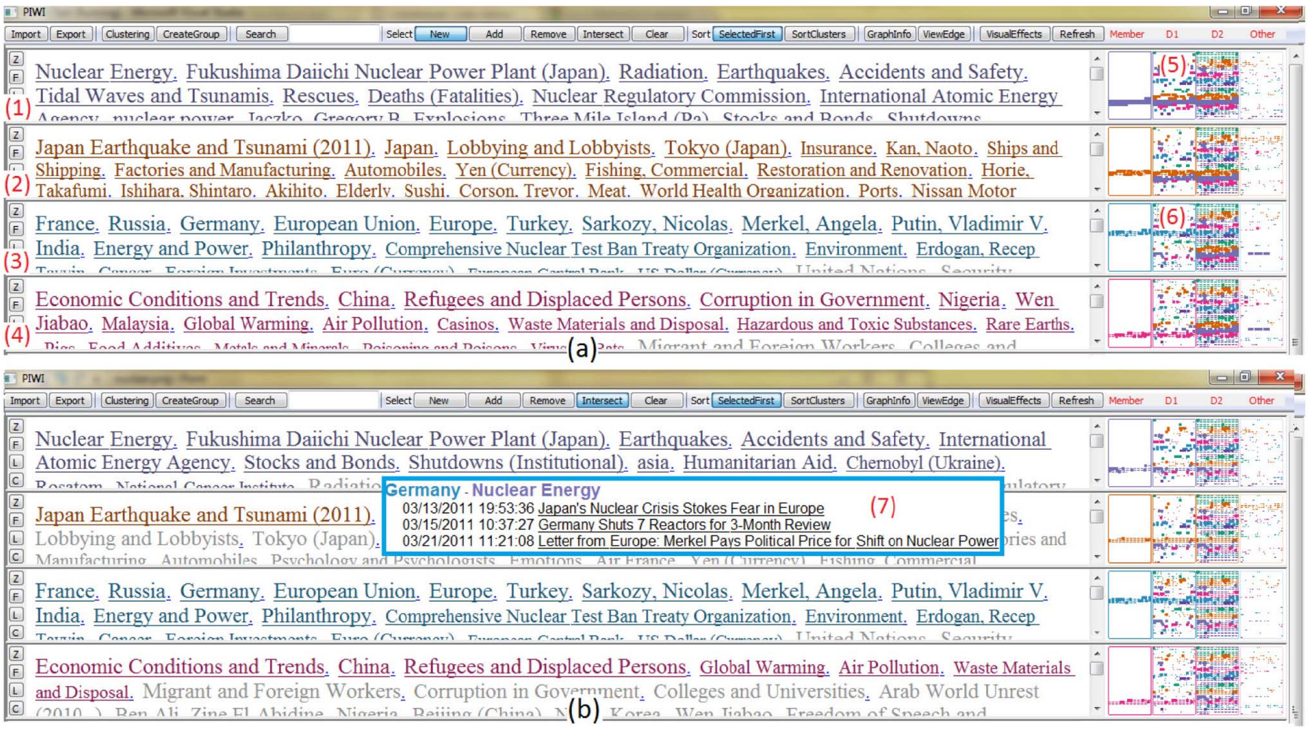
example, the degree centrality can be used if vertices with large numbers of neighbors are of interest, while the betweenness centrality can be used if vertices essential for communicating are of interest. The tags are sorted by the same metric in descending order. To save space, the tag cloud is placed in a scrollable html window. Labels of the vertices with the highest metric values are visible in the initial view and users can access other labels by scrolling the window. A label can appear in multiple tag clouds if the vertex belongs to multiple communities.

Example 1.1 Browsing communities. A user browses the communities (Fig. 1) to get an overview of the NYT graph. This is an easy task since she can read the labels of each community following her daily reading manner. She finds that this corpus contains news about *Middle East and North Africa Unrest (2010-)*, *Japan Earthquake and Tsunami (2011)*, *Fukushima Daiichi Nuclear Power Plant (Japan)*, *Terrorism*, etc.

Vertex plots. PIWI uses vertex plots to display the neighborhood information of the communities. A **vertex plot** displays vertices with desired features as colored dots without overlap. To coordinate the many vertex plots in PIWI, each vertex occupies the same position in all vertex plots. A dot is drawn in that position if the vertex has the attribute/structural feature specified in a plot. Otherwise the position is left blank. The dots are placed line by line from top to bottom, in the order from large communities to small communities. Vertices belonging to multiple communities are placed within the communities they are most related to.

As shown in Fig. 1, each row, representing a community, contains multiple vertex plots. The first plot (from left to right) displays all vertices belonging to the community and is named the member plot. The second plot displays all vertices directly connected to any vertex in the community and is named the Degree-1 neighborhood (D1) plot. The third plot displays vertices whose shortest distances to the community is less than or equal to 2. It is named the Degree-2 neighborhood (D2) plot. Similarly, Degree-3 and larger neighborhoods can be defined. Users interactively set the number of neighborhoods to be displayed according to their exploration tasks. The last plot displays vertices that are not displayed in other plots in the row and is named the "Other" plot.

Example 1.2. Comparing community detection algorithms. A graph mining expert wants to compare the results of multiple community detection algorithms. Through the vertex plots displayed in Fig. 1, she notices that the communities resulting from the AdjCluster algorithm [38] have similar sizes (according to the member plots)



Nuclear Energy. Fukushima Daiichi Nuclear Power Plant (Japan). Radiation. Earthquakes. Accidents and Safety. Tidal Waves and Tsunamis. Rescues. Deaths (Fatalities). Nuclear Regulatory Commission. International Atomic Energy Agency. nuclear power. Jaczko, Gregory B. Explosions. Three Mile Island (Pa). Stocks and Bonds. Shutdowns

Japan Earthquake and Tsunami (2011). Japan. Lobbying and Lobbyists. Tokyo (Japan). Insurance. Kan, Naoto. Ships and Shipping. Factories and Manufacturing. Automobiles. Yen (Currency). Fishing, Commercial. Restoration and Renovation. Horie, Takafumi. Ishihara, Shintaro. Akihito. Elderly. Sushi. Corson, Trevor. Meat. World Health Organization. Ports. Nissan Motor

France. Russia. Germany. European Union. Europe. Turkey. Sarkozy, Nicolas. Merkel, Angela. Putin, Vladimir V. India. Energy and Power. Philanthropy. Comprehensive Nuclear Test Ban Treaty Organization. Environment. Erdogan, Recep Tayyip. Cancer. Foreign Investments. Euro (Currency). European Central Bank. US Dollar (Currency). United Nations. Security

Economic Conditions and Trends. China. Refugees and Displaced Persons. Corruption in Government. Nigeria. Wen Jiabao. Malaysia. Global Warming. Air Pollution. Casinos. Waste Materials and Disposal. Hazardous and Toxic Substances. Rare Earths. Pigs. Food Additives. Metals and Minerals. Poisoning and Poisons. Virus Pets. Migrant and Foreign Workers. Colleges and

Nuclear Energy. Fukushima Daiichi Nuclear Power Plant (Japan). Earthquakes. Accidents and Safety. International Atomic Energy Agency. Stocks and Bonds. Shutdowns (Institutional). asia. Humanitarian Aid. Chernobyl (Ukraine). Rosatom. National Cancer Institute. Radiation ... ulatory

Japan Earthquake and Tsunami (2011). ... Lobbying and Lobbyists. Tokyo (Japan). ... ries and Manufacturing. Automobiles. Psychology and Psychologists. Emotions. Air France. Yen (Currency). Fishing, Commercial

Germany - Nuclear Energy
03/13/2011 19:53:36 Japan's Nuclear Crisis Stokes Fear in Europe
03/15/2011 10:37:27 Germany Shuts 7 Reactors for 3-Month Review
03/21/2011 11:21:08 Letter from Europe: Merkel Pays Political Price for Shift on Nuclear Power

France. Russia. Germany. European Union. Europe. Turkey. Sarkozy, Nicolas. Merkel, Angela. Putin, Vladimir V. India. Energy and Power. Philanthropy. Comprehensive Nuclear Test Ban Treaty Organization. Environment. Erdogan, Recep Tayyip. Cancer. Foreign Investments. Euro (Currency). European Central Bank. US Dollar (Currency). United Nations. Security

Economic Conditions and Trends. China. Refugees and Displaced Persons. Global Warming. Air Pollution. Waste Materials and Disposal. Migrant and Foreign Workers. Corruption in Government. Colleges and Universities. Arab World Unrest (2010-). Ben Ali, Zine El Abidine. Nigeria. Beijing (China). N... Korea. Wen Jiabao. Freedom of Speech and

Fig. 2. (a) (1) is a community of interest and (2), (3), and (4) are communities closely related to it. (b) Vertices connecting (1) and (3) are selected by intersecting vertices displayed in (5) and (6). The selected vertices are displayed in bigger dots than unselected vertices in the vertex plots. Their labels are underscored in the tag clouds. The labels of unselected vertices are grayed out.

and each community can connect to most vertices in the graph in two steps (according to the D2 plots). She reclusters the graph using the MCL algorithm [34] and examines the new vertex plots (please watch the supplementary video, available online, for the new display). She finds that the new results contain communities with much more variation in size and connectivity.

3.2 Interactions

Interactions play an essential role in PIWI. Users can effectively select vertices with desired structural features and vertex attributes for complex graph analysis tasks. Progressive visual exploration can be conducted by iteratively constructing User-Defined Vertex Groups (U-groups).

3.2.1 Selection

PIWI maintains a **selection set**. Vertices in this set are called selected vertices. The selection set is interactively modified by users through mouse clicks. We use the term **hits** to refer to all vertices displayed in a vertex plot a user just clicked or a vertex whose label was just clicked. According to the selection status, which is defined by the users through the toolbar, hits are used to modify the selection set as follows:

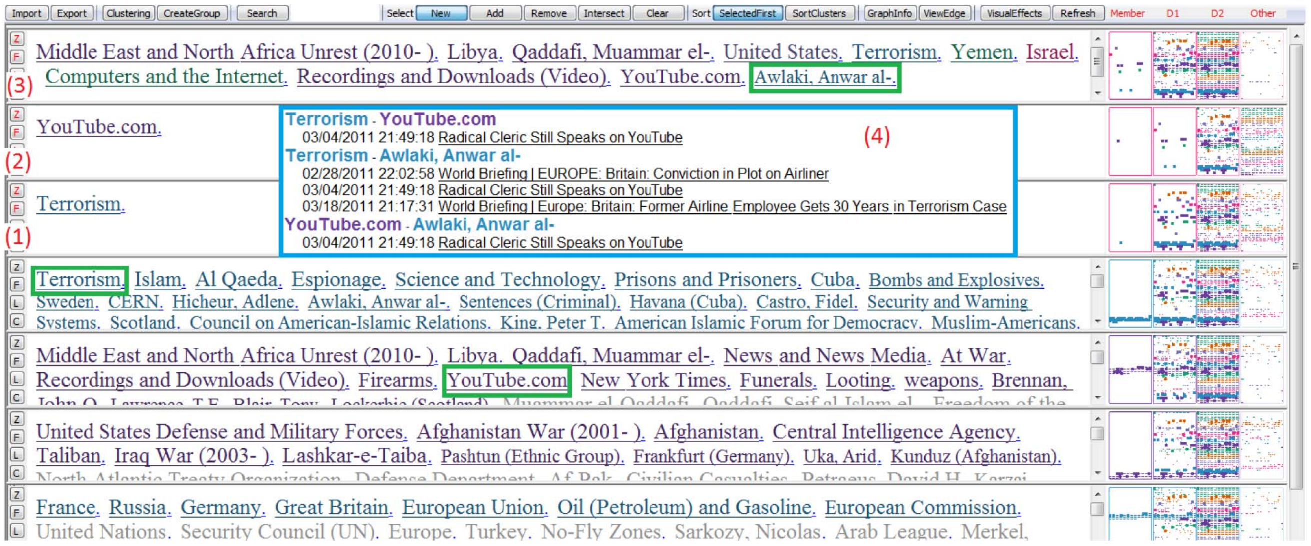
1. "New": the selection set is cleared and the hits are added into the set;
2. "Add": the hits are added into the selection set;
3. "Remove": the hits are removed from the selection set;
4. "Intersect": vertices not in the hits are removed from the selection set. Users can click the "Clear" button to clear the selection set.

Complex selection tasks can be conducted by a sequence of clicks. For example, intersecting two vertex plots, namely to click one plot under the "New" status and then click another plot under the "Intersect" status, selects all vertices displayed in both plots. It can be used for many purposes: intersecting the D1 plots of two communities selects all vertices connecting to both communities; intersecting the D1 plot of community A with the member plot of community B selects all vertices in community B that connect to community A; intersecting the member plots of two communities selects the vertices shared by these communities. For another example, users can create a union of two communities by clicking the member plot of one community under the "New" status and then clicking the member plot of the other community under the "Add" status. Since PIWI allows users to select a large number of vertices with desired features simultaneously, we say that PIWI supports **bulk selection**.

Selected vertices are highlighted in all displays. In the tag clouds, labels of selected vertices are highlighted by underscores. To further emphasize them, users can 1) gray out the labels of unselected vertices (see Fig. 2b); 2) place selected vertices before unselected vertices in the tag clouds (see Fig. 2); and 3) hide unselected vertices. In the vertex plots, selected vertices are displayed in bigger dots than unselected vertices. Users can manually adjust the sizes for both selected and unselected dots. The users set the above options through the visual effect setting dialog.

Sort clusters. Selected vertices may be distributed in multiple communities. To effectively examine them, users can click the "Sort Clusters" button on the toolbar to bring the communities with a large number of selected vertices to the top of the display.

View edge. To investigate the relationship among the selected vertices, users can click the "View Edge" button on the toolbar. A pop-up window will display all edges connecting the selected vertices. In this example application,



Middle East and North Africa Unrest (2010-). Libya. Qaddafi, Muammar el-. United States. Terrorism. Yemen. Israel. Computers and the Internet. Recordings and Downloads (Video). YouTube.com. Awlaki, Anwar al-.

YouTube.com.

Terrorism.

Terrorism - YouTube.com
03/04/2011 21:49:18 Radical Cleric Still Speaks on YouTube
Terrorism - Awlaki, Anwar al-
02/28/2011 22:02:58 World Briefing | EUROPE: Britain: Conviction in Plot on Airliner
03/04/2011 21:49:18 Radical Cleric Still Speaks on YouTube
03/18/2011 21:17:31 World Briefing | Europe: Britain: Former Airline Employee Gets 30 Years in Terrorism Case
YouTube.com - Awlaki, Anwar al-
03/04/2011 21:49:18 Radical Cleric Still Speaks on YouTube

Terrorism. Islam. Al Qaeda. Espionage. Science and Technology. Prisons and Prisoners. Cuba. Bombs and Explosives. Sweden. CERN. Hicheur, Adlene. Awlaki, Anwar al-. Sentences (Criminal). Havana (Cuba). Castro, Fidel. Security and Warning Systems. Scotland. Council on American-Islamic Relations. King, Peter T. American Islamic Forum for Democracy. Muslim-Americans.

Middle East and North Africa Unrest (2010-). Libya. Qaddafi, Muammar el-. News and News Media. At War. Recordings and Downloads (Video). Firearms. YouTube.com. New York Times. Funerals. Looting. weapons. Brennan,

United States Defense and Military Forces. Afghanistan War (2001-). Afghanistan. Central Intelligence Agency. Taliban. Iraq War (2003-). Lashkar-e-Taiba. Pashtun (Ethnic Group). Frankfurt (Germany). Uka, Arid. Kunduz (Afghanistan).

France. Russia. Germany. Great Britain. European Union. Oil (Petroleum) and Gasoline. European Commission. United Nations. Security Council (UN). Europe. Turkey. No-Fly Zones. Sarkozy, Nicolas. Arab League. Merkel,

Fig. 3. A progressive visual exploration on terrorism events. (1), (2), and (3) are U-groups interactively created. Vertices connecting to *Terrorism* are highlighted. (4) shows the edges among the three vertices highlighted in green boxes.

the titles of the articles where the two tags co-occur are listed under each edge. Users can access the original articles by clicking the titles.

Example 1.3. Investigating the relationship between a community and other communities. A community in the NYT graph with vertices *Nuclear Energy* and *Fukushima Daiichi Nuclear Power Plant (Japan)* triggers the interest of a user (Fig. 2 (1)). She wants to examine how this community is related to other communities. To do so, she sets the selection status to "New," and clicks the D1 plot of this community (Fig. 2 (5)). Now all vertices connecting to this community are selected. She clicks the "Sort Cluster" button to sort the communities by the number of selected vertices they contain in descending order. Now the communities most relevant to the community of interest are brought to the top of the display (Fig. 2a).

The user notices a relevant community with vertices *France*, *Russia*, and *Germany* (Fig. 2 (3)). How is this community related to the community of interest? She sets the selection status to "Intersect" and clicks the D1 plot of this community (Fig. 2 (6)). Now only vertices connecting to both communities are selected, as shown in Fig. 2b. Some of them belong to other communities. To focus on the selected vertices in these two communities, the user removes vertices in other communities from the selection set (set status to "Remove" and click the member plots of other communities) and clicks the "View Edge" button. From the html window, she finds interesting articles such as one titled *Japan's nuclear crisis strokes fear in Europe* (Fig. 2 (7)). She clicks the titles to read the articles to get a better understanding of the relationship.

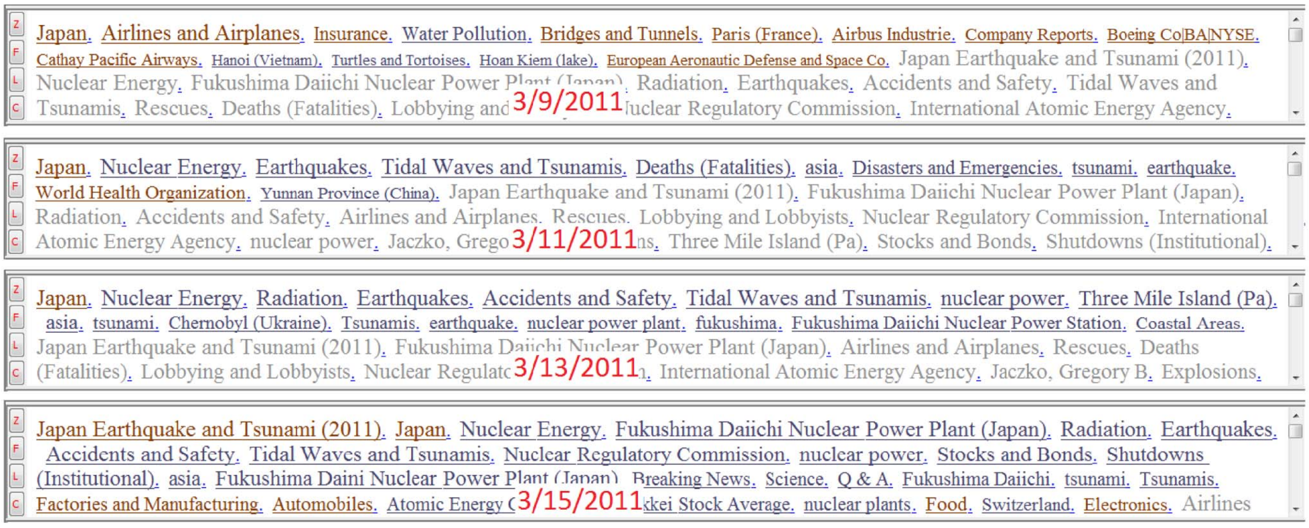
3.2.2 U-Groups

U-groups are user defined vertex clusters. They are visualized and interactively explored in the same way as automatically detected communities. To distinguish U-groups from automatically detected communities, the former are displayed on top of the latter and their left buttons are red (see Fig. 3 (1), (2), and (3)).

To create a U-group, users first select vertices of interest and then click the "Create Group" button from the toolbar. A new U-group will be created which contains all the selected vertices when the button is clicked. Users can create any number of U-groups and delete a U-group at any time. A vertex can be included in multiple U-groups and communities and hence allows users to examine its context from different viewpoints.

Example 1.4. Investigating vertices of interest using U-groups. A user is interested in terrorism events reported in the NYT corpus. When browsing the communities, she notices a community with the vertex *Terrorism*. She wants to examine vertices connecting to *Terrorism* within and outside this community. To do so, she selects this vertex by clicking its label and creates a U-group for it (Fig. 3 (1)). By clicking the D1 plot of this U-group, all vertices connecting to *Terrorism* are selected and highlighted. She sorts the communities by the number of selected vertices they contain, as shown in Fig. 3. It is not surprising that many vertices from other communities are also connected to *Terrorism*. When browsing these vertices, she finds an interesting vertex *Youtube.com*. Why is *Youtube.com* related to *Terrorism*? She decides to examine the common neighbors of these two vertices. She creates a U-group for *Youtube.com* (Fig. 3 (2)). Then, she intersects the D1 plots of the U-groups (1) and (2). Now, the common neighbors of *Terrorism* and *Youtube.-com* are selected. To examine the selected vertices easier, she creates another U-group (Fig. 3 (3)) to accommodate them. The tag *Awlaki, Anwar al-* triggers her interest. She selects this vertex together with *Terrorism* and *Youtube.com* and examines the edges among them using the "View Edge" function (see Fig. 3 (4)). The news article *Radical Cleric Still Speaks on YouTube* gives her the answer why these vertices are related.

In the above example, a progressive visual exploration has been conducted. Intermediate results are recorded in the U-groups, which can be kept for future use or be removed if they are of no further interest.


Z F L C Japan. Airlines and Airplanes. Insurance. Water Pollution. Bridges and Tunnels. Paris (France). Airbus Industrie. Company Reports. Boeing Co|BA|NYSE. Cathay Pacific Airways. Hanoi (Vietnam). Turtles and Tortoises. Hoan Kiem (lake). European Aeronautic Defense and Space Co. Japan Earthquake and Tsunami (2011). Nuclear Energy. Fukushima Daiichi Nuclear Power Plant (Japan). Radiation. Earthquakes. Accidents and Safety. Tidal Waves and Tsunamis. Rescues. Deaths (Fatalities). Lobbying and 3/9/2011 luclear Regulatory Commission. International Atomic Energy Agency.

Z F L C Japan. Nuclear Energy. Earthquakes. Tidal Waves and Tsunamis. Deaths (Fatalities). asia. Disasters and Emergencies. tsunami. earthquake. World Health Organization. Yunnan Province (China). Japan Earthquake and Tsunami (2011). Fukushima Daiichi Nuclear Power Plant (Japan). Radiation. Accidents and Safety. Airlines and Airplanes. Rescues. Lobbying and Lobbyists. Nuclear Regulatory Commission. International Atomic Energy Agency. nuclear power. Jaczko, Grego 3/11/2011 ns. Three Mile Island (Pa). Stocks and Bonds. Shutdowns (Institutional).

Z F L C Japan. Nuclear Energy. Radiation. Earthquakes. Accidents and Safety. Tidal Waves and Tsunamis. nuclear power. Three Mile Island (Pa). asia. tsunami. Chernobyl (Ukraine). Tsunamis. earthquake. nuclear power plant. fukushima. Fukushima Daiichi Nuclear Power Station. Coastal Areas. Japan Earthquake and Tsunami (2011). Fukushima Daiichi Nuclear Power Plant (Japan). Airlines and Airplanes. Rescues. Deaths (Fatalities). Lobbying and Lobbyists. Nuclear Regulatc 3/13/2011 n. International Atomic Energy Agency. Jaczko, Gregory B. Explosions.

Z F L C Japan Earthquake and Tsunami (2011). Japan. Nuclear Energy. Fukushima Daiichi Nuclear Power Plant (Japan). Radiation. Earthquakes. Accidents and Safety. Tidal Waves and Tsunamis. Nuclear Regulatory Commission. nuclear power. Stocks and Bonds. Shutdowns (Institutional). asia. Fukushima Daini Nuclear Power Plant (Japan). Breaking News. Science. Q & A. Fukushima Daiichi. tsunami. Tsunamis. Factories and Manufacturing. Automobiles. Atomic Energy (3/15/2011 kkei Stock Average. nuclear plants. Food. Switzerland. Electronics. Airlines

Fig. 4. A U-group which is a union of (1) and (2) in Fig. 2. The vertices highlighted by underscores were used to tag one or more NYT news articles on the date marked.

Example 1.5. Merging closely related communities. The two communities in Fig. 2 (1) and (2) are closely related. The user decides to examine them together. She selects both communities and creates a new U-group to accommodate them (see Fig. 4).

3.2.3 Other Interactions

Import: Users can load a graph with or without community detection results. If community detection results are not provided, PIWI will cluster the graph automatically using the community detection algorithm selected by the user.

Export (Save subgraphs): Users can save the subgraph consisting of vertices in the selection set and those edges connecting them as a new graph. The new graph can be loaded into PIWI for further analysis.

Clustering: Users can select a different community detection algorithm and then recluster the graph.

Search: PIWI allows users to search vertices containing text entered in the "Search" box on the toolbar. To make the results visible, the vertices will be selected and a U-group will be created to hold them.

Sort: Besides sorting communities by the number of selected vertices they contain, users can manually move a community to the top, to the bottom, or remove it from the display by clicking the small "F," "L," or "C" button on the left of a row.

Zoom: Users can click the "Z" button on the left of a row to examine its labels in a large pop-up html window.

3.3 Visualization of Multivariate Graphs

Vertex plots can represent vertex attributes as well as structural information. A binary attribute is represented by a vertex plot that displays vertices whose values are 1. Categorical attributes with N distinct values are represented by N vertex plots, each of which displays all vertices with a distinct value. Numeric attributes are discretized and then visualized. Bulk selections can be conducted on vertex plots representing vertex attributes in the same way as on vertex plots representing neighborhoods.

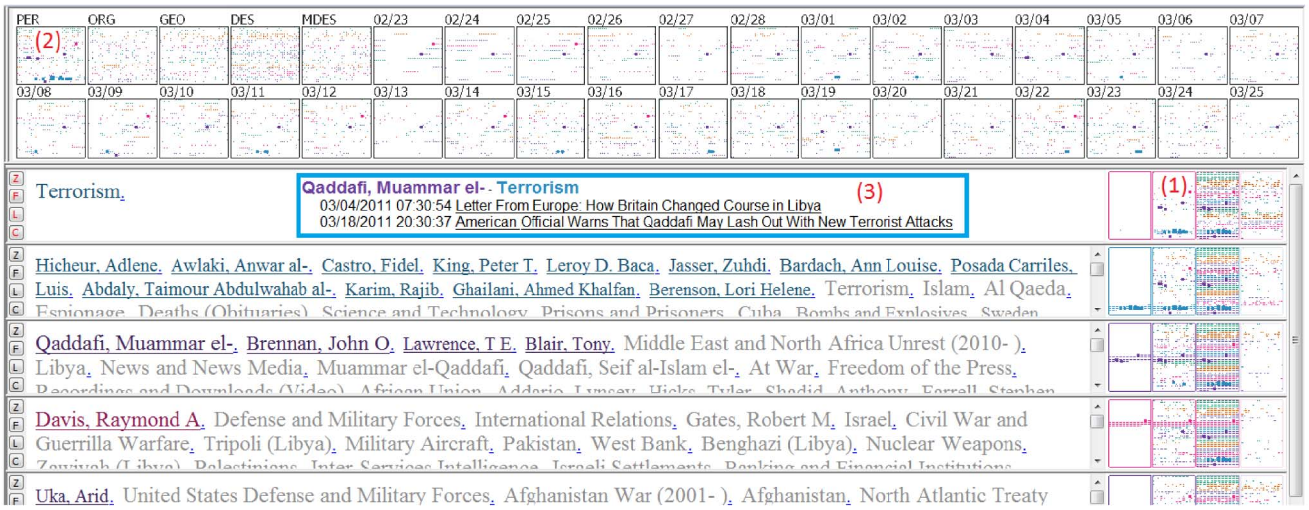
Example 1.6. Exploring multivariate vertex attributes. Vertices in the NYT graph carry 36 binary attributes, as shown on the top of Fig. 5. Five of them record the categories of the vertices, such as *PER* (person), *ORG* (organization), and *GEO* (geospatial information). Each of the other 31 attributes is a time stamp used to record whether a vertex was used to tag any NYT articles published on the date it represents.

First, a user wants to find vertices in the person category that are related to *Terrorism*. To do so, she creates a U-group for *Terrorism* and intersects its D1 plot (Fig. 5 (1)) with the *PER* vertex plot (Fig. 5 (2)). Now all vertices in the person category connecting to *Terrorism* are selected. These vertices and their context are brought to the top of the display after the user clicks the "Sort Cluster" button (see Fig. 5).

Second, the user wants to investigate how the Japan earthquake was reported by the New York Times. To do so, she clicks the vertex plots of the time stamps (see Fig. 5) one by one under the "New" status heading to examine how vertices are selected in the U-group she created in Example 1.5. Fig. 4 shows a few screenshots she sees. The progress of how this news event was reported by NYT is clearly revealed in this way.

4 Case Study

In this case study, a user investigates the InfoVis coauthor network which has 1,462 vertices (authors) and 6,075 edges (coauthorships). The network was generated using 4,526 bibliography entries from the DBLP Computer Science Bibliography [1] in 2011. All of them are authored/coauthored by 36 researchers who served on the program committee of the IEEE conference on Information Visualization. From this network, the user wants to identify significant author communities and explore the relationships among them. She also wants to identify collaboration leaders and investigate their collaboration patterns. *The details of this case study can be seen from the supplementary video,* available online.



Terrorism.

Qaddafi, Muammar el- - Terrorism
03/04/2011 07:30:54 Letter From Europe: How Britain Changed Course in Libya
03/18/2011 20:30:37 American Official Warns That Qaddafi May Lash Out With New Terrorist Attacks

Hicheur, Adlene. Awlaki, Anwar al-. Castro, Fidel. King, Peter T. Leroy D. Baca. Jasser, Zuhdi. Bardach, Ann Louise. Posada Carriles, Luis. Abdaly, Taimour Abdulwahab al-. Karim, Rajib. Ghailani, Ahmed Khalfan. Berenson, Lori Helene. Terrorism. Islam. Al Qaeda. Espionage. Deaths (Obituaries). Science and Technology. Prisons and Prisoners. Cuba. Bombs and Explosives. Sweden.

Qaddafi, Muammar el-. Brennan, John O. Lawrence, T E. Blair, Tony. Middle East and North Africa Unrest (2010-). Libya. News and News Media. Muammar el-Qaddafi. Qaddafi, Seif al-Islam el-. At War. Freedom of the Press. Recordings and Downloads (Video). African Union. Addario, Lynsey. Hicks, Tyler. Shadid, Anthony. Farrell, Stephen.

Davis, Raymond A. Defense and Military Forces. International Relations. Gates, Robert M. Israel. Civil War and Guerrilla Warfare. Tripoli (Libya). Military Aircraft. Pakistan. West Bank. Benghazi (Libya). Nuclear Weapons. Zawiyah (Libya). Palestinians. Inter-Services Intelligence. Israeli Settlements. Banking and Financial Institutions.

Uka, Arid. United States Defense and Military Forces. Afghanistan War (2001-). Afghanistan. North Atlantic Treaty

Fig. 5. The highlighted vertices are persons related to *Terrorism* in the NYT graph. They are selected by intersecting (1) and (2). (3) shows the articles connecting *Quddafi, Muammar, el* and *Terrorism*.

Example 2.1. Browsing communities. The user loads this network into PIWI and clusters it using the Clauset-Newman-Moore algorithm [12]. Twenty-one communities are generated and visualized. Fig. 6 shows the most significant communities. Collaboration leaders in these communities, whose names are displayed in large fonts on top of the tag clouds, are easily identified.

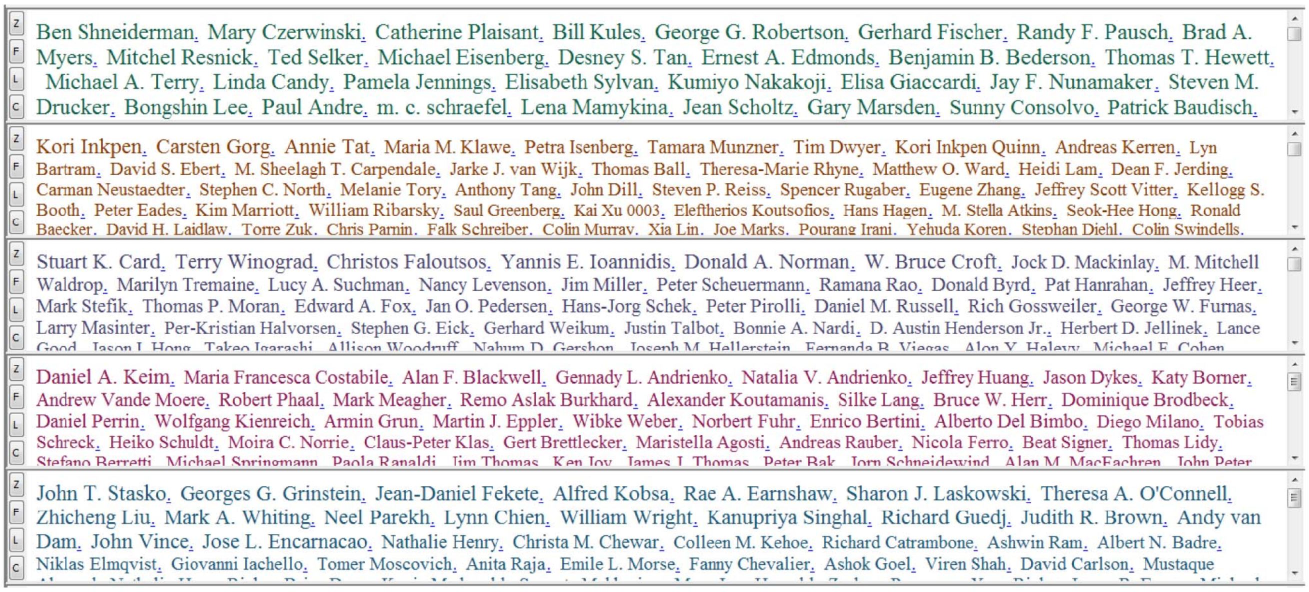
Example 2.2. Finding disconnected communities. The user is curious to know if there is any community having no collaborations with the biggest community. She selects its neighbors by clicking its D1 plot and sorts the communities by the number of selected vertices they contain. She then drags the scrolling bar to the bottom. At the bottom she finds three communities containing no selected vertices, which indicates that these communities are disconnected from the large one at the top.

Example 2.3. Finding bridging vertices. The user wonders if there are any researchers in other communities who bridge the biggest community with a community disconnected to it. She intersects the D1 plots of these two communities. The selected vertices are the possible bridging persons.

Example 2.4. Investigating the relationship between a vertex and multiple communities. The user wants to know the collaboration patterns of a research leader *John T. Stasko*. She creates a U-group for *John T. Stasko* and clicks its D1 plot to select his collaborators. She sorts the communities by the number of selected vertices they contain. She immediately notices that there are six communities besides John's own community whose members collaborate with John.

5 IMPLEMENTATION

A fully working prototype of PIWI has been implemented in C++, using wxWidgets [6] for its GUI and SNAP [4],



Ben Shneiderman. Mary Czerwinski. Catherine Plaisant. Bill Kules. George G. Robertson. Gerhard Fischer. Randy F. Pausch. Brad A. Myers. Mitchel Resnick. Ted Selker. Michael Eisenberg. Desney S. Tan. Ernest A. Edmonds. Benjamin B. Bederson. Thomas T. Hewett. Michael A. Terry. Linda Candy. Pamela Jennings. Elisabeth Sylvan. Kumiyo Nakakoji. Elisa Giaccardi. Jay F. Nunamaker. Steven M. Drucker. Bongshin Lee. Paul Andre. m. c. schraefel. Lena Mamykina. Jean Scholtz. Gary Marsden. Sunny Consolvo. Patrick Baudisch.

Kori Inkpen. Carsten Gorg. Annie Tat. Maria M. Klawe. Petra Isenberg. Tamara Munzner. Tim Dwyer. Kori Inkpen Quinn. Andreas Kerren. Lyn Bartram. David S. Ebert. M. Sheelagh T. Carpendale. Jarke J. van Wijk. Thomas Ball. Theresa-Marie Rhyne. Matthew O. Ward. Heidi Lam. Dean F. Jerding. Carman Neustaedter. Stephen C. North. Melanie Tory. Anthony Tang. John Dill. Steven P. Reiss. Spencer Rugaber. Eugene Zhang. Jeffrey Scott Vitter. Kellogg S. Booth. Peter Eades. Kim Marriott. William Ribarsky. Saul Greenberg. Kai Xu 0003. Eleftherios Koutsofios. Hans Hagen. M. Stella Atkins. Seok-Hee Hong. Ronald Baecker. David H. Laidlaw. Torre Zuk. Chris Parnin. Falk Schreiber. Colin Murray. Xia Lin. Joe Marks. Pourang Irani. Yehuda Koren. Stephan Diehl. Colin Swindells.

Stuart K. Card. Terry Winograd. Christos Faloutsos. Yannis E. Ioannidis. Donald A. Norman. W. Bruce Croft. Jock D. Mackinlay. M. Mitchell Waldrop. Marilyn Tremaine. Lucy A. Suchman. Nancy Levenson. Jim Miller. Peter Scheuermann. Ramana Rao. Donald Byrd. Pat Hanrahan. Jeffrey Heer. Mark Stefik. Thomas P. Moran. Edward A. Fox. Jan O. Pedersen. Hans-Jorg Schek. Peter Pirolli. Daniel M. Russell. Rich Gossweiler. George W. Furnas. Larry Masinter. Per-Kristian Halvorsen. Stephen G. Eick. Gerhard Weikum. Justin Talbot. Bonnie A. Nardi. D. Austin Henderson Jr.. Herbert D. Jellinek. Lance Good. Jason I. Hong. Takeo Igarashi. Allison Woodruff. Nahum D. Gershon. Joseph M. Hellerstein. Fernanda B. Viegas. Alon Y. Halevy. Michael F. Cohen.

Daniel A. Keim. Maria Francesca Costabile. Alan F. Blackwell. Gennady L. Andrienko. Natalia V. Andrienko. Jeffrey Huang. Jason Dykes. Katy Borner. Andrew Vande Moere. Robert Phaal. Mark Meagher. Remo Aslak Burkhard. Alexander Koutamanis. Silke Lang. Bruce W. Herr. Dominique Brodbeck. Daniel Perrin. Wolfgang Kienreich. Armin Grun. Martin J. Eppler. Wibke Weber. Norbert Fuhr. Enrico Bertini. Alberto Del Bimbo. Diego Milano. Tobias Schreck. Heiko Schuldt. Moira C. Norrie. Claus-Peter Klas. Gert Brettlecker. Maristella Agosti. Andreas Rauber. Nicola Ferro. Beat Signer. Thomas Lidy. Stefano Berretti. Michael Springmann. Paola Ranaldi. Jim Thomas. Ken Joy. James J. Thomas. Peter Bak. Jorn Schneidewind. Alan M. MacEachren. John Peter

John T. Stasko. Georges G. Grinstein. Jean-Daniel Fekete. Alfred Kobsa. Rae A. Earnshaw. Sharon J. Laskowski. Theresa A. O'Connell. Zhicheng Liu. Mark A. Whiting. Neel Parekh. Lynn Chien. William Wright. Kanupriya Singhal. Richard Guedj. Judith R. Brown. Andy van Dam. John Vince. Jose L. Encarnacao. Nathalie Henry. Christa M. Chewar. Colleen M. Kehoe. Richard Catrambone. Ashwin Ram. Albert N. Badre. Niklas Elmqvist. Giovanni Iachello. Tomer Moscovich. Anita Raja. Emile L. Morse. Fanny Chevalier. Ashok Goel. Viren Shah. David Carlson. Mustaque

Fig. 6. The most significant communities in the Infovis co-author network (1,462 vertices, 6,075 edges).

igraph [2], and MCL [34] for graph analysis. The tag clouds are displayed in html. The dots in the vertex plots are drawn using OpenGL as GL points.

For graphs with several thousand vertices, such as the example graphs in this paper, PIWI provides uncluttered display and real-time interactions on an average consumer PC or laptop without any acceleration techniques. When visualizing even larger graphs, PIWI needs special handling in the implementation to reduce clutter and accelerate the rendering. Since the tag clouds are displayed using scrollable html windows, they never get cluttered. To make the rendering faster, PIWI provides the option to limit the number of labels displayed in each tag cloud. The other labels are displayed upon request.

A typical vertex plot in PIWI contains about 10,000 pixels (100×100). Thus PIWI can scale to graphs with about 10,000 vertices without clutter (each dot is represented by one pixel, and the selected and unselected dots are distinguished by using different transparencies). Users can make this number larger by reducing the number of vertex plots in each row, by decreasing the width of tag clouds, or by increasing the height of rows.

Rendering acceleration with GPU: When rendering a graph with N vertices and k vertex plots with a CPU, $k \times N$ GL points need to be drawn one by one. When N is over several thousand, vertex plot rendering becomes a bottleneck for interactive exploration. To address this problem, a GPU accelerated version using CG shaders is implemented. In particular, an inverse pixel-by-pixel technique is used: a determination is made for each pixel in the vertex plot as to whether it belongs to any dot. After determining which dot the pixel belongs to, the pixel is assigned a color. GPU textures are used to store necessary rendering parameters, including *{relative position—vertex id} mapping*, *{vertex id—community id} mapping*, *{community id—color} mapping*, *{vertex id—binary attribute} mapping*, *{vertex id—distance to community/U-group} mapping*, *{vertex id—highlight state} mapping*, etc. All calculations of pixel positions, colors, show states, and highlight states are then moved from CPU calculations to GPU texture look-up. The capability of parallel pixel processing of the GPU fragment shader program largely improves the speed of vertex plot rendering, while maintaining exactly the same rendering results. The acceleration is especially beneficial when the graph is large and the pixel number assigned to each vertex is small.

A set of rendering tests have been conducted to evaluate the effectiveness of the acceleration method. Eight coauthor networks collected from the DBLP Computer Science Bibliography were used (see Table 1 for their sizes). All tests were executed on a Dell T3500 Workstation with a 2.66 GHz Intel E7300 CPU, 4 GB memory, and an NVidia GTX 570 graphics card with 1,280 MB video memory. In the experiments, the CPU version and the GPU version rendered each graph 100 times, respectively, to get their average time for refreshing all vertex plots. It turned out that the GPU version accelerated the rendering approximately 1 to 3.6 times. Detailed comparison results are shown in Table 1. The acceleration was negligible when the graph size was small (less than 1,000 vertices). When the graphs became larger, the speedup increased and reached its maximum for a graph with 10,000 vertices. The speedup slightly decreased for even larger graphs, but was still significant for graphs with up to 100,000 vertices. When graphs have more than 100,000 vertices, the vertex plots will be cluttered anyway and vertex plots need to be replaced by other visual metaphors.

TABLE 1
The GPU Acceleration Speedup

$N(G)$	$E(G)$	$N(NP)$	$T_C(s)$	$T_G(s)$	$Speedup$
500	1,704	5	0.017	0.016	1.06
1,000	3,927	12	0.038	0.021	1.81
2,000	8,167	22	0.092	0.032	2.88
5,000	23,355	51	0.451	0.125	3.61
10,000	53,477	51	0.945	0.260	3.63
20,000	110,339	51	2.669	0.836	3.19
50,000	291,482	51	14.110	5.458	2.59
100,000	583,319	51	57.420	25.230	2.28

$N(G)$ and $E(G)$ are graph vertex counts and edge counts, respectively; $N(NP)$ is the number of vertex plots rendered; $T_C(s)$ and $T_G(s)$ are CPU and GPU rendering time in seconds, respectively; and $Speedup$ is the speedup of GPU rendering over CPU rendering.

6 User Study

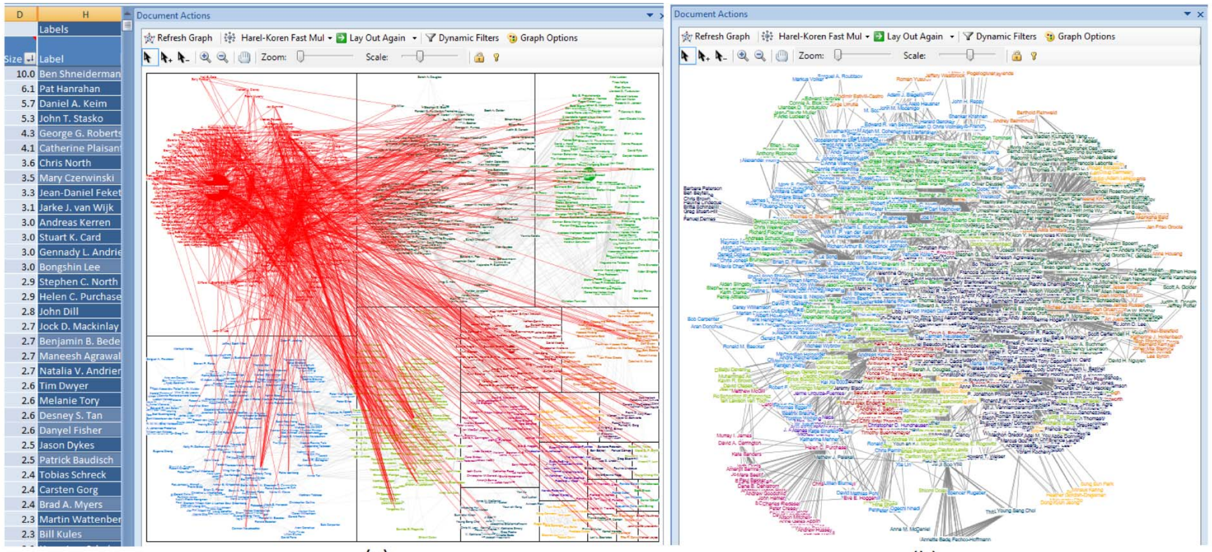
6.1 Setup

A preliminary user study has been conducted to evaluate the effectiveness of PIWI in helping sophisticated users conducting community related tasks on a real graph. In this study, PIWI (CPU version) was compared with the state-of-the-art graph visualization tool NodeXL (version 1.0.1.196 released on December 2011, the newest downloadable version when the study was conducted in February 2012) [3]. NodeXL displayed graphs whose edge and vertex lists were stored in an Excel 2007 or Excel 2010 workbook (Excel 2007 was used in the study). It uses coordinated NLDs and Excel worksheets to display the graphs and their vertex, edge, and group details.

Subjects: Ten subjects participated in this study. They were graduate and senior undergraduate students of CS and other engineering majors. They were not familiar with PIWI and NodeXL, although they all knew NLDs. The subjects were divided into two groups balanced by the number of subjects, major, and education level. One group used PIWI and the other group used NodeXL in the study.

Data sets and their communities: Two graphs were used in the study. The Polbooks graph (105 vertices and 441 edges) [33] was used in the training. The Infovis coauthor Network (1,462 vertices and 6,075 edges) was used in the test. Four communities were detected from the Polbooks graph and twenty-one communities were detected from the Infovis coauthor network using the Clauset-Newman-Moore algorithm [12]. The same community detection results were used in NodeXL and PIWI.

NodeXL setup: 1) NLD layout: each of a graph's community was laid out in a box. The boxes were sorted by community size (see Fig. 7a). This layout was chosen since it was the best layout for community related analysis tasks in NodeXL (Figs. 7a and 7b compare the layout used and another layout in NodeXL). 2) Visual effects: Vertex size helped the subjects identify vertices with high-degree



Size	Label
10.0	Ben Shneiderman
6.1	Pat Hanrahan
5.7	Daniel A. Keim
5.3	John T. Stasko
4.3	George G. Roberts
4.1	Catherine Plaisant
3.6	Chris North
3.5	Mary Czerwinski
3.3	Jean-Daniel Feket
3.1	Jarke J. van Wijk
3.0	Andreas Kerren
3.0	Stuart K. Card
3.0	Gennady L. Andrie
3.0	Bongshin Lee
2.9	Stephen C. North
2.9	Helen C. Purchase
2.8	John Dill
2.7	Jock D. Mackinlay
2.7	Benjamin B. Bede
2.7	Maneesh Agrawal
2.7	Natalia V. Andrie
2.6	Tim Dwyer
2.6	Melanie Tory
2.6	Desney S. Tan
2.6	Danyel Fisher
2.5	Jason Dykes
2.5	Patrick Baudisch
2.4	Tobias Schreck
2.4	Carsten Gorg
2.4	Brad A. Myers
2.3	Martin Wattenber
2.3	Bill Kules

Fig. 7. NLDs in NodeXL [3]. The Infovis coauthor network is displayed. (a) The box layout used in the user study. Vertices in the top left community are selected. (b) The Harel-Koren fast multiscale layout.

centralities. The edges were set to be semitransparent to reduce clutter. The subjects were allowed to change the opacity interactively during the study.

NodeXL interactions: The subjects were trained to use the following interactions in NodeXL: 1) Selection: Selecting vertices would highlight them and all edges connecting to them in both the NLD and the Excel worksheets. Clicking a vertex on the NLD or the Excel worksheet would select it. Drawing a rectangle on the NLD would select all vertices in it. This technique was used to select all vertices in a community. Fig. 7a shows the results of such a selection. A right click on a vertex would trigger a menu from which all neighbors of the vertex could be selected. Users could add/remove newly selected vertices to/from the selection set, but intersection was not available in NodeXL. 2) Clutter reduction: Dynamic filters would be used to remove vertices with small degree centralities from the NLD. Scaling the vertices and labels and zooming in would reduce overlap among the labels. In addition, if a vertex was selected from the NodeXL, it would be automatically brought to the center of the Excel worksheet, where its label could be read feasibly. 3) Search: Excel search functions could be used to find vertices whose labels contained a given string. There were more interactions available in NodeXL, but they were not relevant to this study.

Procedure: The subjects conducted the study one by one on a Lenovo T420 laptop (Intel Core i5-2410M CPU@2.30 GHZ). The laptop was connected to an external display set with a 20 inch LCD monitor with a resolution of $1{,}680 \times 1{,}050$ pixels. Each subject had a training session followed by a test session. In the training session, the instructor first briefly introduced the visualization and the interactions of the system to a subject. Then, the subject conducted four training tasks on the Polbooks graph, which were highly similar to the tasks to be conducted on the Infovis co-author graph in the test session. Hands-on help was provided by the instructor, including exemplifying strategies for conducting the task and answering questions. The training session ended when the subject finished the training tasks successfully. The total time spent in the training session was recorded for each subject. In the test session, the subject conducted four tasks on the Infovis coauthor graph independently. The time she/he spent on each task was recorded. After each task, the subject was asked to rate the effectiveness of the tool on a five point scale. After the test session, the subjects provided preference ratings and comments.

The four tasks used in the test are presented as follows. They were significant community related analysis tasks identified from the case study (see Section 4). The best strategies for using NodeXL to conduct most tasks, figured out by the authors in the pilot study and taught to the subjects in the training, were also presented. The strategies taught to the PIWI users are presented in Examples 2.1, 2.2, 2.3, and 2.4 in Section 4 and are not repeated here. The subjects were free to use any other strategies they wanted. In all tasks the subjects only needed to write down the first names of the authors to reduce the effect of their writing speed on performance.

Task 1: Browsing communities: Identify four big communities. From each of them, find three persons with high degree centralities and write down their first names. **NodeXL Strategy:** Apply the dynamic filter to remove vertices with small degree centralities. Scale the labels and zoom into big boxes for labels of large vertices. **Alternative NodeXL strategy:** Sort the vertices in the Excel worksheet by their degree centrality in descending order. Select all vertices of a community displayed in a large box. Copy the labels of the three topmost selected vertices from the worksheet. This was a more efficient strategy than the first one for large graphs. The first strategy, rather than the second one, was taught to the subjects to test the capability of the subjects in reading labels from an NLD.

Task 2: Finding disconnected communities: Given the label of a vertex (*Ben Shneiderman*), find three groups that are not directly connected to his community. Write down the first names of three persons from each group. **NodeXL strategy:** Search and select the given vertex. Identify his community and select all vertices in this community. The

TABLE 2
User Study Results

		PIWI	NodeXL
Task 1	Time (minutes)	1.8	4.7
	Correctness (%)	100	100
	Effectiveness Rating (1-5)	5	3.6
Task 2	Time (minutes)	2.7	4.9
	Correctness (%)	100	47
	Effectiveness Rating (1-5)	4.8	2.6
Task 3	Time (minutes)	2.6	5.6
	Correctness (%)	100	33
	Effectiveness Rating (1-5)	4.6	2.2
Task 4	Time (minutes)	1.9	3.7
	Correctness (%)	100	97
	Effectiveness Rating (1-5)	5	2.8
Easy to Use (1-5)		3.2	2.2

vertices and all edges connecting to them are thus highlighted (see Fig. 7a). Scan the NLD for communities where there is no vertex connecting to the highlighted edges.

Task 3: Finding bridging vertices: Labels of two vertices are given (*Helen C. Purchase* and *Ben Shneiderman*). Their communities are not directly connected. Are there any persons in other groups who connect these two communities? If so, write down their first names. The answer had three bridging vertices. **NodeXL strategy:** Select vertices in both communities and manually look for bridging vertices.

Task 4: Investigating the relationship between a vertex and multiple communities There are six communities connecting to *John T. Stasko* besides his own community. Write down the first name of a person connecting to *John T. Stasko* from each of these communities. **NodeXL Strategy:** Select the vertex and its adjacent vertices. Look for selected vertices from different boxes.

Please note that NodeXL is a general purpose graph visualization system, while PIWI is specially designed for community related tasks. All the tasks used in the study were community related tasks and thus they might favor the design features of PIWI. Future studies will have to be conducted to understand which tasks are convenient with PIWI and which favor other tools in a wider scope.

6.2 Results

The average training time was about the same for NodeXL (9.7 minutes) and PIWI (9.3 minutes). The average completion time and correctness of PIWI users and NodeXL users on each task are reported in Table 2. After each task, the subjects answered the question "Is this tool effective in conducting this task?" on a five point scale (5-strongly agree, 1-strongly disagree). After all tasks, the subjects also gave preference ratings on "easy to use" (5-very easy, 1-very difficult). The average ratings from PIWI users and NodeXL users are reported in Table 2. As shown in Table 2, the five PIWI users had faster average performance times than the five NodeXL users for all tasks. PIWI users also had higher average correctness than NodeXL users for tasks 2, 3, and 4. PIWI users thought their tool was more effective than NodeXL users for all four tasks. In general, PIWI users thought their tool was easier to use than NodeXL users. We expect that if the alternative strategy was taught in Task 1, the NodeXL users would have had better performance in Task 1. The current result of Task 1 revealed the difficulty of label reading in NLDs if no other facilitated views were provided.

After the test, the subjects wrote down their comments on the advantages and disadvantages of the system they used. PIWI users listed the following advantages: "I felt that the system was well organized, which made detail operations really easy to do." "Easy for tasks. Enable me to quickly access the connected vertices without cluttering the display." "This system is excellent for searching through and determining relationships within graph data." "Easy to find neighbors of a given vertex and common neighbors of several vertices." "I like the vertex plots for selection." "The ability to quickly construct new groups gave a lot of additional information on the relation between these groups." "The sort cluster function was very helpful." All of the disadvantages listed for PIWI were mild suggestions, such as adding a search group function and adding a right click menu to avoid having to move to the toolbar when changing the selection status. These suggestions will be applied to the new version of PIWI.

NodeXL received positive comments such as the interactions were simple, community relationships were revealed to some extent, the integration with Excel was good, and the highlighting and selection were helpful. The negative comments revealed that multiple NodeXL users experienced difficulties in selection: "Sometimes I need to click vertices one by one to find the answers. Crazy." "Difficult to select vertices due to the label clutter." "This system is easy to understand, but not effective. I have to search over the interface to find answers." The subjects also complained about the clutter in the display: "Labels are overlapped. Hard to recognize." "When a vertex has too many neighbors, it is hard to distinguish them."

7 Exploratory Study for Utility

In the user study reported in Section 6, the subjects conducted graph analysis tasks following given strategies. To learn how PIWI was used in a more complex visual reasoning process, another exploratory study was conducted with sophisticated users. In this study, eight computer science graduate students were asked to use PIWI (CPU version) to gather information from the NYT news corpus introduced in Section 3. Two tasks were given to each subject. The first task was to identify as many news events about Libya as possible in 10 minutes. The second task was to identify three terrorism events in the shortest time possible. For each subject, the instructor first showed her/him how to use PIWI to explore the NYT graph, identify vertices and edges of interest, and access the original news articles through the edges. The subject then tested the interactions with the help of the instructor. This free form exploration lasted about 10 minutes. After that, she/he conducted the tasks independently. After conducting the tasks, she/he rated PIWI in "usefulness," "ease of use," and "awareness of context" on five point scales and answered open questions.

On average, 3.75 news events were identified by the subjects in the first task. The subjects completed the second task in 7.5 minutes on average. It was observed that the subjects developed different strategies to make use of PIWI.

For example, some of them first browsed the communities in the initial view and then created U-groups for vertices of interest; some of them started by searching vertices of interest and then examined communities most relevant to these vertices; some of them started by bulk selection (such as selecting all vertices connecting to terrorism), followed by exporting and exploring subgraphs. These strategies showed that PIWI was a flexible graph exploration tool.

With regard to preference ratings, PIWI received average ratings of 4, 3.75, and 4.38 on "usefulness," "ease to use," and "awareness of context," respectively, (1 is the lowest and 5 is the highest). The poststudy comments from the subjects showed that they liked the vertex plots. There were comments such as "The vertex plot is powerful. I can quickly erase the unnecessary vertices. I like it" and "The vertex plot can help me find the relationship among different types of information." The subjects also commented that using the vertex plots for selection was much quicker and intuitive than using menus or check lists. In addition, the subjects commented that the tag clouds made sense to them. There were comments such as "Perfect. The tag clouds provide direct description of what I am working at!"

8 Discussion

8.1 Contributions

The large amount of vertices and edges as well as the complexity of their inner structure hinder immediate knowledge discovery in graphs. Since the display and human perception ability are limited, we carefully select the information to be presented in the initial view. An iterative, progressive knowledge discovery process is facilitated by interactions in support of the

1. definition of interesting vertex sets;
2. identification of neighbors/paths;
3. investigation of detailed contents; and
4. acknowledgement of global context.

Tag clouds and vertex plots (they can be viewed as a variant of the Pixel-Oriented techniques [23]) are familiar visualization techniques. PIWI creatively assembles them to carry semantic, structural, and attribute information of a graph. Thus, a graph is untangled into a set of communities that can be displayed row by row in a scrollable window, which greatly expands the visualization space and capability. Consequently, plenty of space is assigned to vertex labels. Vertex attributes can also be explored together with the structural information seamlessly. This method loses direct links between graph vertices. However, the links are a major source of clutter and typically play a negative role in visualizing and analyzing graphs with more than hundreds of vertices. More importantly, the interactive exploration through bulk selections and U-groups compensates the loss and promotes effective investigation in exploratory tasks. This satisfied the needs of real users in the evaluation.

Bulk selections and U-groups are novel, useful interactions for making sense of large graphs. Bulk selections allow users to select vertices using complex attribute and structural criteria simultaneously with ease. With U-groups, the cognitive load of users can be reduced by allowing complex graph exploration tasks to be conducted within several steps of simple operations. U-groups also enable users to explore a large graph progressively and iteratively.

If the vertex plots of a vertex group (a community or a U-group) are considered a visual presentation of the distances from all vertices to that group, PIWI actually maps a graph from a high-dimensional space to a much lower dimensional space. This space consists of dimensions that summarize the structural information (communities and their neighbors), dimensions that reflect user interests (U-groups and their neighbors), and vertex attributes.

8.2 Applicability and Limitations

Currently, PIWI supports visual exploration of unweighted, undirected graphs with thousands of vertices where the semantics of the vertex labels matter. PIWI can also present binary attributes carried by the vertices and support the visual exploration of relationships between these attributes and graph structure. For graphs where vertices are merely numbered, edges have many attributes, or vertices have multiple continuous attribute values, other strategies may be better. PIWI is yet to be evaluated on a wider range of graph analysis tasks, including both community related tasks and noncommunity related tasks.

PIWI has the potential to be applied to a wider range of graphs. It can be extended to visualize directed graphs by displaying incoming neighbors and outgoing neighbors in different vertex plots. Graphs with weighted edges would be visualized by displaying vertices within different weighted distance ranges to a community in multiple vertex plots. An alternative solution is to replace the vertex plots by other visual metaphors, such as dynamic query scales [29], to display the weighted distances in a continuous scale. To visualize multimodal graphs, vertices of different types may need to be separated into different groups. More interactions for exploring relationships among groups of different types may need to be provided. In addition, adopting visual metaphors such as dynamic query scales [29] could allow PIWI to better support visualization of graphs with numeric vertex attributes.

PIWI is a useful tool for sophisticated users, as shown in the user studies. Further user studies need to be conducted to test its usability for users without an engineering background. In case that it is too complex for novice users, the visualization can be simplified to enhance usability at the cost of the amount of information presented. For example, vertex plots can be simplified into glyphs indicating the populations and the percentages of vertices in the neighborhood that have been selected.

PIWI can visualize communities constructed based on graph structure, vertex attributes, or any other information. By interactively creating U-groups, users can even utilize and compare different ways of grouping. For example, they can construct U-groups for vertices with different attributes and compare them with those communities constructed using structure based clustering. They can also intersect vertex plots representing communities and attributes to construct new groups with both attributes and structural information considered.

The effectiveness of the initial views in PIWI depends on the quality of the community detection algorithm used.

However, this reliance is not considered a vital issue in PIWI since 1) Community detection is a well-studied field and many algorithms have been proven to be very effective in real graphs. PIWI can easily integrate many algorithms. 2) The dependency of visualization on clustering quality exists in most clustering-based visualization approaches. PIWI actually has less such dependency than many other approaches. The reason is that the progressive visual reasoning supported by PIWI allows users to conduct further analysis after an initial insight is discovered rather than making decisions upon possibly misleading insights.

Furthermore, this weakness can be alleviated along the following directions: 1) allowing users to interactively control the clustering process and monitor the clustering quality through metrics such as modularity and visual exploration; 2) allowing users to visually compare the clustering results from different algorithms to find the optimal ones. This function is important since an algorithm working well for certain types of graphs may have poor performance on other types of graphs.

9 Conclusion and Future Work

In this paper, we propose PIWI, a novel visualization system that allows users to interactively explore large multivariate graphs based on their community structure. Its "tag clouds + vertex plots" visualization presents the semantics, attributes, and communities of a large graph in an uncluttered way. Labels of a large number of vertices can be examined within their communities and neighborhood context simultaneously. PIWI provides bulk selections to facilitate the selection of a large number of vertices according to desired structural features and vertex attributes. Moreover, user-defined vertex groups, which can be constructed efficiently with bulk selections, allow users to use the output of a previous step as the input to further visual exploration, and then to compare, refine, and integrate results from multiple visual exploration steps. Our case studies and user studies demonstrated that PIWI facilitated rich community related graph analysis tasks and iterative visual reasoning. The GPU accelerated version of PIWI scales to graphs with ten thousands of vertices. In the future, we will extend PIWI for other types of graphs. Another important task is to systematically study community related graph analysis tasks by constructing a task taxonomy and exploring the design space of visual analytics approaches to these tasks.

Acknowledgments

The work is partially supported by the US National Science Foundation (NSF) under Grant Number IIS-0946400, SBE-0915528, and IIS-0916131, National Natural Science Foundation of China Project (No. 60903062 and 61170204), and the DHS Visual Analytics for Command, Control, and Interoperability (VACCINE) Center of Excellence, under the auspices of the SouthEast Regional Visual Analytics Center. The authors would like to thank Dr. Xintao Wu and Leting Wu for their help in community analysis and the anonymous reviewers of *TVCG* for their valuable suggestions.

References

[1] The DBLP Computer Science Bibliography. http://www.informatik.uni-trier.de/ley/db/, 2012.
[2] The Igraph Library for Complex Network zresearch, http://igraph.sourceforge.net, 2012.
[3] NodeXL: Network Overview, Discovery and Exploration for Excel, http://nodexl.codeplex.com/, 2012.
[4] SNAP: Stanford Network Analysis Project. http://snap.stanford.edu/, 2012.
[5] The New York Times, http://www.nytimes.com, 2012.
[6] wxWidgets: A cross-platform GUI library. http://www.wxwidgets.org/, 2012.
[7] J. Abello and F. van Ham, "Matrix Zoom: A Visual Interface to Semi-External Graphs," *Proc. IEEE Symp. Information Visualization,* pp 183-190, 2004.
[8] A. Aris and B. Shneiderman, "Designing Semantic Substrates for Visual Network Exploration," *Information Visualization,* vol. 6, no. 4, pp. 281-300, 2007.
[9] P. Bausch and J. Bumgardner, *Make a Flickr-Style Tag Cloud,* second ed., O'Reilly Press 2006.
[10] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps.* Univ. of Wisconsin Press, 1983.
[11] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete, "GraphDice: A System for Exploring Multivariate Social Networks," *IEEE Trans. Visualization and Computer Graphics,* vol. 29, no. 3, pp. 1100-1108, Aug. 2010.
[12] A. Clauset, M.E.J. Newman, and C. Moore, "Finding Community Structure in Very Large Networks," *Physical Rev. E,* vol. 70, no. 6, 066111, 2004.
[13] J. Ellson, E.R. Gansner, E. Koutsofios, S.C. North, and G. Woodhull, "Graphviz and Dynagraph - Static and Dynamic Graph Drawing Tools," *Graph Drawing Software,* M. Junger and P. Mutzel, eds., pp. 127-148, Springer-Verlag, 2003.
[14] S. Fortunato, "Community Detection in Graphs," *Physics Reports,* vol. 486, no. 3, pp. 75-174, 2010.
[15] Y. Fua, M. Ward, and E. Rundensteiner, "Hierarchical Parallel Coordinates for Exploration of Large Datasets," *Proc. IEEE Visualization,* pp. 43-50, Oct. 1999.
[16] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the Readability of Graphs Using Node-Link and Matrix-Based Representations: A Controlled Experiment and Statistical Analysis," *Information Visualization,* vol. 4, no. 2, pp. 114-135, 2005.
[17] M. Girvan and M. Newman, "Community Structure in Social and Biological Networks," *Proc. Nat'l Academy of Sciences of the United States of Am.,* vol. 99, no. 12, pp. 7821-7826, 2002.
[18] M.A. Hearst and J.O. Pedersen, "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results," *Proc. 19th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 76-84, 1996.
[19] J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks," *Proc. IEEE Symp. Information Visualization,* pp. 5-12, 2005.
[20] N. Henry and J.-D. Fekete, "MatrixExplorer: A Dual-Representation System to Explore Social Networks," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 667-684, Sept./Oct. 2006.
[21] N. Henry, J.-D. Fekete, and M.J. McGuffin, "NodeTrix: A Hybrid Visualization of Social Networks," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1302-1309, Nov./Dec. 2007.
[22] T. Itoh, C. Muelder, K.-L. Ma, and J. Sese, "A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs," *Proc. IEEE Pacific Visualization Symp.,* pp. 121-128, 2009.
[23] D. Keim and H.-P. Kriegel, "VisDB: Database Exploration Using Multidimensional Visualization," *IEEE Computer Graphics and Applications,* vol. 14, no. 5, pp. 40-49, Sept. 1994.
[24] M.J. McGuffin and I. Jurisica, "Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 6, pp. 937-1944, Nov./Dec. 2009.
[25] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," *Nature,* vol. 435, no. 7043, pp. 814-818, 2005.
[26] A. Perer and B. Shneiderman, "Balancing Systematic and Flexible Exploration of Social Networks," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 639-700, Sept./Oct. 2006.
[27] A. Pretorius and J. van Wijk, "Visual Analysis of Multivariate State Transition Graphs," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 685-692, Sept./Oct. 2006.

[28] Z. Shen, K.-L. Ma, and T. Eliassi-Rad, "Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 6, pp. 1427-1439, Nov./Dec. 2006.

[29] B. Shneiderman, "Dynamic Queries for Visual Information Seeking," *IEEE Software,* vol. 11, no. 6, pp. 70-77, Nov. 1994.

[30] B. Shneiderman and A. Aris, "Network Visualization by Semantic Substrates," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 733-740, Sept./Oct. 2006.

[31] A. Singla and I. Weber, "Camera Brand Congruence in the Flickr Social Graph," *Proc. Second ACM Int'l Conf. Web Search and Data Mining (WSDM '09),* pp. 252-261, 2009.

[32] J.T. Stasko, C. Gorg, and Z. Liu, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization," *Information Visualization,* vol. 7, no. 2, pp. 118-132, 2008.

[33] V. Krebs, "The Polbooks Dataset," http://www.orgnet.com/, 2012.

[34] S. van Dongen, "Graph Clustering by Flow Simulation," PhD thesis, Univ. of Utrecht, 2000.

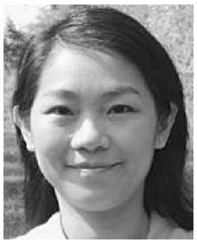
[35] F. van Ham and A. Perer, "'Search, Show Context, Expand on Demand': Supporting Large Graph Exploration with Degree-of-Interest," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 6, pp. 953-960, Nov./Dec. 2009.

[36] C. Viau, M.J. McGuffin, Y. Chiricota, and I. Jurisica, "The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 6, pp. 1100-1108, Nov./Dec. 2010.

[37] M. Wattenberg, "Visual Exploration of Multivariate Graphs," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems,* pp. 811-819, 2006.

[38] L. Wu, X. Ying, X. Wu, and Z.-H. Zhou, "Line Orthogonality in Adjacency Eigenspace with Application to Community Partition," *Proc. Int'l Jojnt Conf. Artificial Intelligence (IJCAI),* pp. 2349-2354, 2011.

[39] J. Yang, D. Luo, and Y. Liu, "Newdle: Interactive Visual Exploration of Large Online News Collections," *IEEE Computer Graphics and Applications,* vol. 30, no. 5, pp. 32-41, Sept./Oct. 2010.



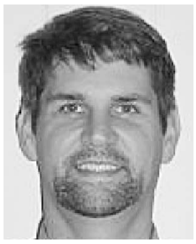
Jing Yang received the PhD degree in computer science from Worcester Polytechnic Institute. She is an associate professor in the Computer Science Department at the University of North Carolina at Charlotte. Her research interests include visual analytics and information visualization on high-dimensional data, graphs, and text documents.

Yujie Liu received the MS degree in automatic control from Huazhong University of Science and Technology, and is working toward the PhD degree in the Computer Science Department at the University of North Carolina at Charlotte. Her research interests include evaluations and visual analytics on graphs and text documents.

Xin Zhang received the BS degree in psychology and the BS degree in computer science from Peking University. He is working toward the master's degree on computer science at School of EECS, Peking University. His research interests lie in information visualization and visual analytics, with emphasis on graph visualization and social network visual analysis.

Xiaoru Yuan received the PhD degree in computer science from the University of Minnesota at Twin Cities. He is a faculty member at Peking University, in the Laboratory of Machine Perception (MOE). His primary research interests lie in the field of visualization and visual analytics, with emphasis on large data visualization, high-dimensional data visualization, graph visualization, and novel visualization user interface. He is a member of the IEEE.

Ye Zhao received the PhD degree in computer science from Stony Brook University. He is an associate professor in the Department of Computer Science at Kent State University. His research interests include natural phenomena modeling, data visualization and visual analytics. He is a member of the IEEE.



Scott Barlowe received the PhD degree from the Computing and Information Systems program at the University of North Carolina at Charlotte. His research interests include visualization, data mining, and bioinformatics.

Shixia Liu received the PhD degree in computer aided design and computer graphics from Tsinghua University. She is a lead researcher in the Internet Graphics Group at Microsoft Research Asia. Her research interest mainly focuses on interactive, visual text analytics and interactive, visual graph analytics. She is a member of the IEEE.