Effective User Interactions for Visual Analytics Tools

Vladimir Guchev

guchev@di.unito.it

Computer Science Department, University of Turin
Tutor: prof. Cristina Gena

Abstract. In the last few decades, there has formed a layer of traditions in the information processing from different fields of science and technology, which includes a variety of standards, techniques and approaches for visual analysis of diverse types and structures of data. Every year, the advanced technologies of human-computer interaction and computer graphics are becoming closer to information-based areas of activity, providing flexible and adaptive solutions in the implementation of user interfaces for information management and decision-making systems that involving the human-analyst. Taking into account the fact that the resulting quality of activity in various areas of high-tech depends on the ability of studying and mastering of complexly structured storages of multidimensional data, the issue of developing effective user interactions for visual analytics tools is becoming increasingly important. This paper pays attention to the problems and possible solutions of effective visual representation and management of static and dynamic graph-based datasets. Particular attention given to ergonomic approaches of data visualization and concepts of its gesture-based handling and control.

Keywords: human-centered computing, user interface design, data visualization, node-link diagram, graph exploration, pen-centric and sketch-based interaction

1 Introduction

Among the wide variety of books and publications, focused on the intersection of CHI and Data Visualization, there is a bias in applied areas, while more fundamental things as the design and ergonomics of the both of interaction patterns and visual symbolic systems are leaving without due attention. Even the developers of specialised software for data analytics, who should be well acquainted with the capabilities of advanced input devices (e.g. touch screens or surface stylus pens), still remain using interactions that are more typical for the WIMP (windows-icons-mouse-pointer) concept with its limitations. So far, the most of visual analytics tools are inside the button paradigm: that is the user actions are typically launched by the pressing a real or virtual but still button, sometimes even with a stylus or a touch surface.

For a human, well familiar with the culture of writing, the poking of points as the base way of user interactions does not seem to be natural. Having advanced input devices and computing systems, which are enough to enable the adaptive and reactive gesture recognition, now is the good time for the reasonable amplification of low-level interactions in order to reduce the number of micro-gestures (e.g. click sequences) at least for visual data analytics software, based on pen-centric input systems. As a good practical example of effective application of sketch-based interactions, it is worth considering the user interfaces for graph exploration and editing. Due to the large number of demanded functions for application to the graph elements, layouts and views, the implementation of user strokes tracing and recognition in the context of an active interaction mode may be an effective solution. The effectiveness of the mentioned above application example may also depend on the readability of the graph. The topic of visual graph or hyper-graph representation is quite complex and requires special study. However, among the approaches to partially ordered graph visualization, it is worth noting the circular layout that has strengths in the both of the convenient form for visual human perception and the flexibility in arrangement methods.

This paper is organized as follows. The “Related Work” section gives a brief review of graph-based data visualization and manipulation techniques. The “Towards Reactive Interacting” section overviews the approaches to the building of gestures design space and the realisation of reactive interactions. The “Towards Flexible Visualizing” section is reviewing the visualization techniques by means of node-link-group diagrams.

2 Related Work

One of the most demanded concepts for visual analytics tools that has a rich theoretical foundation, which allowing to study multidimensional datasets with complex structures, is the interactive graph exploration. The significance and ubiquity of interactive graph exploration are beyond any doubt: it gives wide opportunities for extensive analysing of relationships and dependencies along with patterns and exceptions in complex data such as biological, transport or financial datasets, etc. Among the variety of combinations and hybrids of visualization methods, node-link diagrams on the force-layout basis remain demanded for flexible representation of network data like collaborative, social or communication networks. However, the readability and manageability of this type of diagrams are an open problem: typical solutions (such as geometric compression, semantic abstraction, topologic simplification, etc.) provide to the analyst albeit complete, but fixed and arduous result [1], [2]. The sensemaking of graph-formatted data involves a wide range of tools for interactive and adaptive exploration. Efficiency and clarity of such approaches for visual analysis depend on the compliance between particular tasks and chosen modalities of interactivity [3], [4], [5]. Fully automated algorithms for graph layout cannot provide a complete solution for effective visualization. Several techniques have been proposed for interacting with graphs, in particular, through customized layouts by adjusting of nodes with interactive lenses and sticks, via magnet-based attraction and radial menus [3], [6], [7]. By ordering the chaos of the initial layout, the analyst gains insights into the instant graph changes [8].

Efforts on post-layout enhancement of graph visualization are usually associated with clusterization and multi-view representation [9], [10]. Few menu-based techniques, providing some ways of interaction that typical for graphic editors [6], may be considered as complicated to use. Studying of graph data through free manipulation of nodes, specifically by partial geometrization of arrangement while preserving the context of nodes in a graph topology, requires the ability of quick direct control of elements: that is achievable through the sketch-based techniques [11], [12].

Focusing on the visual part, it is worth noticing a large number of visualization techniques in the literature that represent graphs using node-link diagrams [13], [26], [29]. Among the rich variety of graph drawing techniques, it would be worth to make a choice of the circular layout as one of the easiest for understanding and implementation. There are many examples of using the circular and radial layouts for analysis of various datasets. Depending on the scope of application, the circular and radial visualizations may be focused on different formats and structures of exploring data, thus have great variety in approaches to the graphical design. Burch et al. present the techniques and metaphors based on a radial node-link approach for encoding of time-series relational data [14], as well as for scalable dynamic graphical visualization [15]. Besides the discussion of benefits and drawbacks of the radial visualization, the work also reveal the static and dynamic aspects in node-link representations. The research of Velhow et al. [16] proposes the approach to the exploration of time-varying relational data, presented by large dynamic graphs, where the edges are defined by the polar coordinates instead of Cartesian coordinates. The approach incorporates several interaction techniques to explore dynamic patterns, such as trends and counter-trends. The research of Holten [17] focuses on datasets, containing hierarchical components, presents solutions for visualizing compound graphs based on bundling of adjacency edges. The results of user studies, where hierarchical edge bundles applied for a few available layouts (such as balloon, rooted tree, squarified treemap), show that the radial layout is the most preferable and aesthetically pleasing.

3 Towards Reactive Interacting

Techniques and tools for visual analytics are often represented by a complex and diverse set of using actions. Typically, each action performed by an elongate combination of separate short-term manipulations with branching menus, multiple sliders and numerous buttons. Although, in some cases, it is more convenient to expand the set of manipulations through use of natural for humans stroking and touching sequences. Within context-aware applications, the access to available actions by sketching and drawing gestures could be obvious and convenient. Interactive data visualization that supplemented with tracing and recognition

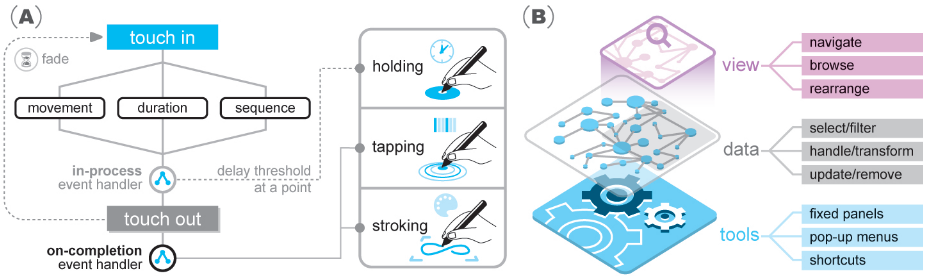

Fig. 1. Types of touch-based actions (A). Architecture of user interface design (B).

of pen-centric sketching, drawing and tapping, which are technically available from any pointing device, opens new opportunities for layout control during exploratory data analysis. The current section presents an approach to gestures processing, comprising a controller that classifies advanced pointer events and a manager that allows handling events on data visualization elements.

3.1 Advanced Pen-centric Control of Data Visualization

In order to expand the range of a user commands set, recognizable by the system, an extended approach to interpreting the input strokes is proposed. To broaden the classification of pen-centric gestures perceived by a system, the set of measurements as in Fig. 1A is proposed for usage. Creating and dynamic updating of the gesture space is available with the potential of reactive programming [18]. Various view-driven actions and data-driven elements can be bind to the preferred tools and functions of interactive visualization: Fig. 1B shows the architecture of user interface along with the related components. In general, the gestures at the visual analytics tool are processed by two possible ways, as shown in Fig. 2A. Depending on the complexity assessment of an input stroke, its' attributes are used for geometric calculations within the frame of related objects positions or as the input vector for recognition by an artificial neural network. For the most efficient distribution of available functionality between handlers, some gestures may be reused for different actions in keeping with conditions of global environment (as it is depicted for zooming levels in Fig. 3).

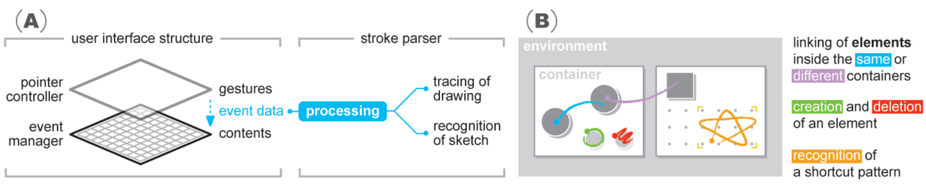

Fig. 2. Concept of pointer events handler (A). Examples of pen-centric actions (B).

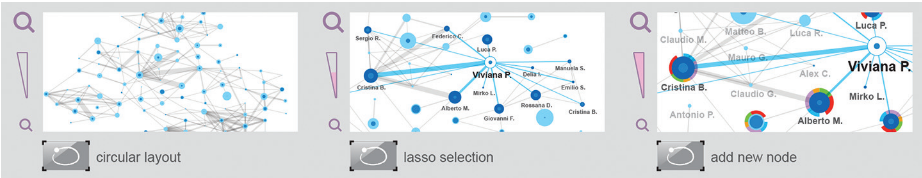

Fig. 3. Reactive adjustment of the system response to a gesture, along with zooming.

3.2 Sketch-based Untangling of Force-directed Graphs

The exploration of moderately dense networks is used in various challenges of visual data analysis. Frequently, the solutions lay in graph drawing, based on automatic force-directed layout, which results in a spontaneous and irreproducible node-link diagram. Currently available approaches to improve its readability are generally oriented to finite rendering without providing to the analyst handy tools for post-layout manipulations. Enabling indirect manual control on visualizations through multi-step menus may appear difficult to learn and use. Thus, the problem requires a more intuitive way of solving. This subsection presents an original toolset for user-guided refinement of the force-directed graph layout, with a bias on sketching techniques. Bearing in mind the ease-of-use and acceptable accuracy of pen-centric manipulations, the sketching technique allow transition of a graph geometry from irregular to regular. The toolset contains simple and intuitive gesture-friendly user interface for view-driven selection, navigation, manipulation, filtering and arrangement of nodes on a graph.

Depending on tasks and goals, performed during data exploration, pen-centric interactions require different levels of input precision. If general draft actions like a preselection (that can be corrected further) or a gesture-command (having a good potential to be recognized) are feasible even with a rough stroke, the manipulations with elements of visualization require greater accuracy. The described user interface solves relatively complex view-driven tasks through easy-performing pointing device gestures of two main types. The first type is the im-

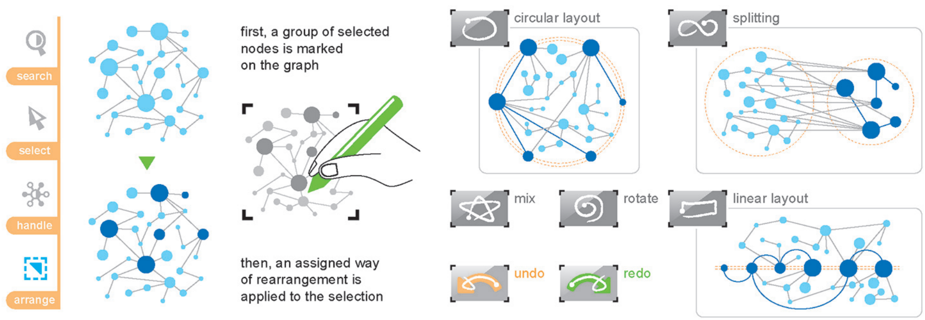

Fig. 4. Triggered by a gesture post-layout manipulations of a selected group of nodes.

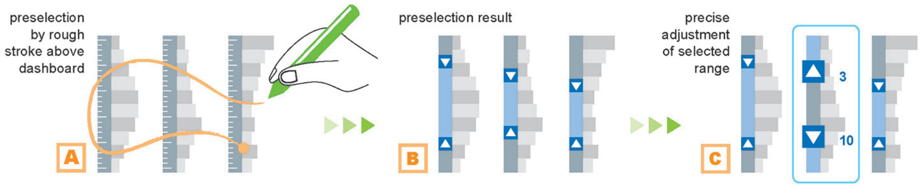

Fig. 5. Scheme of indirect drawing-based selection of a nodes group on a graph. This scenario involves three steps with increasing requirements to the gestures precision.

precise drawing for shape recognition (which applies a corresponding action, in particular the nodes alignment as in Fig. 4) or contextual tracing (for a rough outline of constraints, as on the group selection tool in Fig. 5A). The second type is the precise drag-and-drop manipulation that is a direct moving of graph nodes or an accurate adjustment of the ruler sliders (as in Fig. 5C). The set of pen-based single-stroke gestures for recognition is due to ease of drawing, regardless whether it painted by a hardware stylus, or by a finger on touch-screen device, or by a mouse. Sketching above the graph may be optionally visible: if desired, the user interface allows to enable the indication of gesture path with fading trace of a pointer, as well as to show a recognized pattern upon completion. In addition to the untangling of force-directed layout while stroking above it, the toolset provides usual actions as searching, brushing, picking and direct modification of individual or related nodes and groups.

4 Towards Flexible Visualizing

This section proposes guidelines to provide indications for an effective visual relational analysis and chronology browsing of graph-based (as well as hypergraph-based) datasets. Such guidelines have been defined on the basis of relevant papers and on the study about divergent possible visual representations of hypergraphs. As depicted in Fig. 6A, the semantics of data structure considers:

- Nodes, for instance, members of a collaboration network;
- Links, that are connections between vertices;
- Affiliations, grouping relationships of vertices;
- Relations, the dependencies among existing Groups.

Groups interrelated among themselves, representing a range, a tree, or a sparse network, contain relatively densely linked nodes. Each node may be affiliated with a group or a set of groups. The hyper-edge specifies a set of nodes or an ordered list of nodes. In the following, techniques for representation of nodes, links and groups, along with their composition (see Fig. 6B), are reviewed in order to explore and find proper visualization methods, related to circular, radial or spiral layout for the node-link diagram with groups. This study excludes graphic effects, animation and obvious ways of coding, as colouring and conditional marking [13], [19].

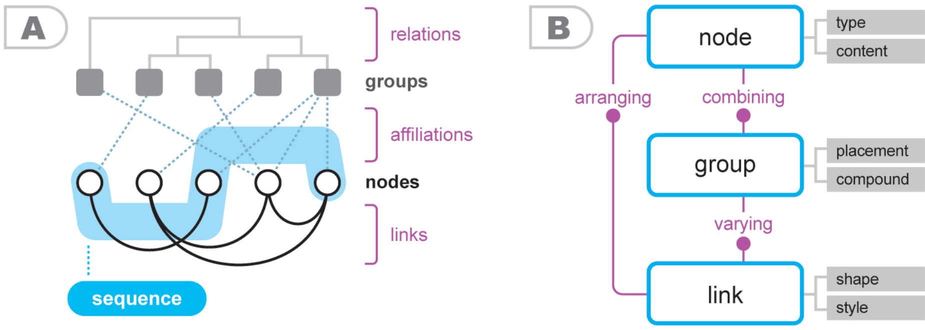

Fig. 6. Data structure considered in the study (A). The scheme of visible elements and the combination of properties, overviewed in the guidelines (B).

4.1 Arrangement of Graph Layout

The displacement of an object on the screen is one of the problems to solve when dealing with graph visualizations. Typically node-link diagrams are shown on a Cartesian plane. There is a number of similar terms describing the items distribution in a circular fashion: circular [20], radial [21] and spiral [22], each referring to a polar coordinates system, where the starting point is the centre of coordinates. The geometric structure of the polar grid may be characterized as comprising two measurements:

- Circular, the position of element along the circular line. Visually, a circular grid is built of concentric circles or a spiral as in Fig. 7B, which can be nested or twisted with increasing radius in arithmetic or geometric progression. If desired, the initially curved lines may be segmented in order to form a set of straightened axes. Depending on the objectives, the curvature of guides may be precluded (see Fig. 7B right) as it is done in the Radar Chart.
- Radial, the distance on a polar axis between element and the centre, i.e. the pole. The main components of the radial grid are guiding lines, diverging from the pole (see Fig 7A). The grid lines can be straight or bowed, with tilt to the rotation axis, starting from centre or indented to periphery. The guides that may contain loops [23] deliberately not considered here.

When the grid consists of concentric circles and centripetal lines and the visualization is crowded, following curved lines can be difficult for a user. In order to ease the reading of adjacent lines or sectors, there are several possible implementations of segmentation of the diagram areas: the sectors, the circles, the zebra stripes segments (Fig. 7D). Visualizations may support extending of a selected sector angle and height stretching of a selected circle (Fig. 7C), or adding of related details (Fig. 7E).

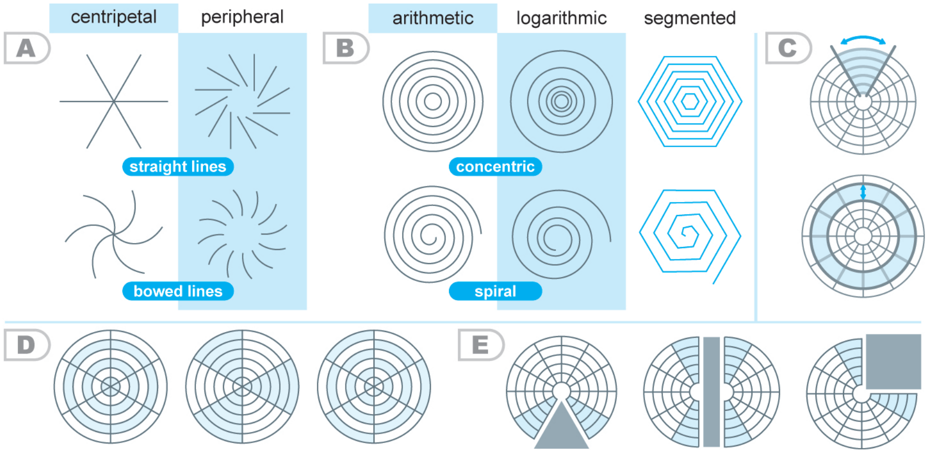

Fig. 7. Grid types in polar coordinates: radial (A), circular and spiral (B). Zoom implementation (C). Basic approaches to grid segmentation (D). Most common solutions for embedding of auxiliary elements (E).

4.2 Basic Composition Elements

The main components of the diagram are nodes, links and groups, which are described below. According to Krzywinski [24], [25], the most common elements for multilayer circular layout of nodes are the following (see Fig. 8A):

- Glyphs: symmetric symbols, miniatures or motifs, having mnemonic extent, used to denote vertices on a graph. The glyph refers to a unique data unit that may be paired with text.
- Patterns: a rectangular shape, transformed in the polar system that may contain statistical graphics like scatter plots and charts.
- Silhouettes: a form of continuous line or area; they convey contextual time-series or quantitative fluctuations. There is a wide variety of positioning for different types of layers (see Fig. 8B).

The purpose of a link between a node pair is the binding of two elements of the same type with a single line. This paragraph intentionally does not mention the styling techniques that use colour, thickness, and other accessible attributes of a line, which are widely observed in surveys and literature [26], [27]; it presents the systematization of geometric variations of the link contours, along with the approach to combining of a guide grid with a coordinate plane for the purpose to indicate quantitative values. The analysis of complex multi-dimensional datasets often requires handling data items of multiple types and also dealing with a rich variety of connecting ways between such data (see Fig. 9C). The main design goal is the achievement of visual distinctiveness between different types of linking. An important aspect of the linking line forming in a graph visualized in Cartesian plane (as in Fig. 9B) is the shaping of lines (as in Fig. 9A).

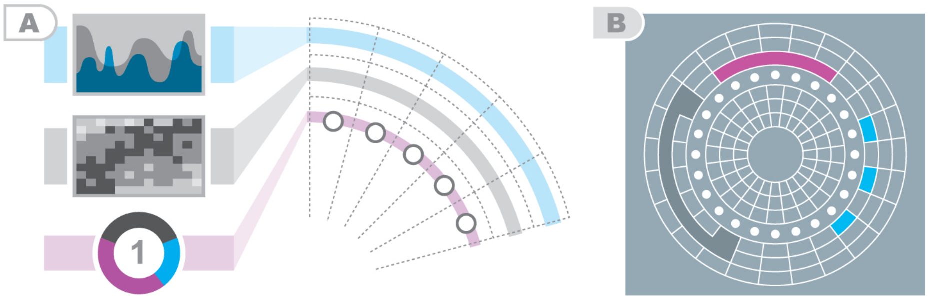

Fig. 8. Basic types of layers for circular layout (A): silhouette, pattern, glyph. Combining variations of layers (B): direct review for the related item (blue), overview for a group of neighbours (violet), report for a pair of distant nodes (dark grey).

Considering that the polar coordinates grid has two basic elements (the circle and the intersection of straight lines) among the variations of the line shape it is worth highlighting the following. *Orthogon*, which, in principle, may represent an alternate polygon fitted for the polar grid properties. Applicable for symmetrically spaced distant nodes. *Round*, which follows the shape of a circular diagram and looks neat at close radial distances. *Conoid*, in contrast to the previous types, it is fully overlaid on the grid. It combines the ability to vary in two dimensions at once, through a rotation angle and a height above/below the node placement guide. In addition to joining a pair of nodes, it may denote the quantitative or chronological properties of a link by the sequence of bends.

Many graphical solutions were proposed and applied to indicate containment contexts of node sets: i.e. belonging of data units to certain classes for both hyper-links in graphs [28] and groups in node-link diagrams [29]. In comparison with probabilistic layout techniques (e.g. springs-embedded [30]), the precise geometric layouts give more possibilities for visualizing containment.

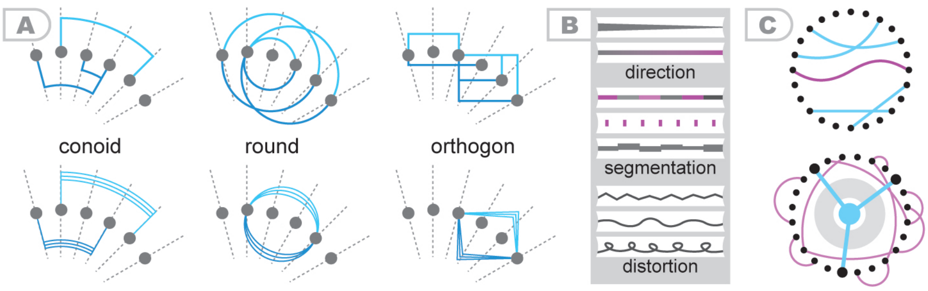

Fig. 9. Regular geometric transformations of linking (A, top) and multi-linking (A, bottom) lines. Modulation of direct linking lines by styling (B). Variations of link styles, depending on a distance between nodes on a circular guide (C).

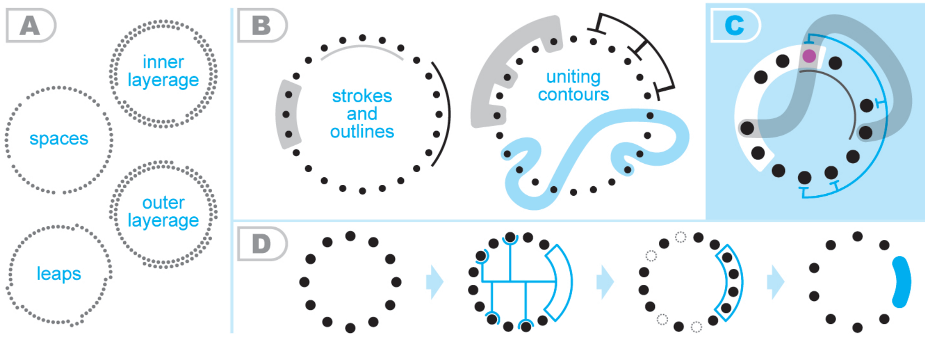

Fig. 10. Grouping through distortion of alignment patterns (A). Visual solutions for neighboring (B, left) and distant (B, right) nodes. Five containment contexts of a violet node via presented techniques (C). Steps of compression of nodes into a group (D).

Visual grouping of nodes can be done using the alignment by a grid (in order to achieve uniformity or proximity of adjacent units) and express containment through graphic connectors. In the case of alignment (Fig. 10A), the visual effect of grouping is achievable through gaps and spaces on the circular visualization, as well as using layers. To express containment, neighbour nodes can be combined in groups of geometric outlines or underlines (Fig. 10B, left), distant nodes can be gathered in groups with compound geometric or free-form contours and paths (Fig. 10B, right), or via visual compression (as in Fig. 10D).

Depending on the task and the level of visual differentiation for node-link-group diagram items, the circular layout may contain combinations of completely different approaches to design or synthesis. Having a rich and wide overview of the variety of representing modes for each element, it is important to know how to combine them effectively and properly for practical applications.

5 Conclusions and Future Work

Definitely, the visual analytics technologies have great potential for increasing the effectiveness of interactions during data exploration, as well as for improving the readability and aesthetic qualities of data visualization.

Among the advantages of drawing gestures is the distribution of application functions between different physical user actions. Implementation of the presented interaction style for enough complex data visualization tools, intended for use by qualified users, can speed up the analytical work by replacing multi-step actions (like selections in cascade menus) with pen-centric shortcuts. During filtering and querying, when it is required just to “outline the boundaries” of interesting values, this interaction style allows to significantly reduce a number of short-term user manipulations with control elements by accomplishing the task in one gesture.

The future work towards reactive interacting will expand the capacity of sketch-based gestures, applicable in visual analysis of various by type and structure datasets, through the usage of multiple-touch and multiple-stroke actions, and their combinations. Particular attention will be paid to studying of the users learnability and preferences in a choice of gesturing methods and techniques for specific view-driven applications.

Developing of the guidelines and framework for graph-based data visualizing using circular techniques, which mainly focuses on the different possibilities of visual representation of graphs and hyper-graphs, has a purpose to help practitioners to have an easy reference in choosing the right technique according to specific needs. This work will continue by assembling existing techniques in the literature and developing the software framework that provides a systematic proposal of a set of techniques, which can be useful for designing of static or dynamic graphs and hyper-graphs.

References

1. Leonel Merino, Dominik Seliner, Mohammad Ghafari, Oscar Nierstrasz. CommunityExplorer: A Framework for Visualizing Collaboration Networks. Proceedings of the 11th Edition of the International Workshop on Smalltalk Technologies (IWST16), Prague, Czech Republic (2016)
2. Xiaoru Yuan, Limei Che, Yifan Hu, Xin Zhang, Intelligent Graph Layout Using Many Users Input. IEEE Trans. on Visualization and Computer Graphics (2012)
3. S. Gladisch, U. Kister, C. Tominski, R. Dachselt, H. Schumann, Mapping Tasks to Interactions for Graph Exploration and Graph Editing on Interactive Surfaces. IEEE Conference on Information Visualization (InfoVis15), Chicago, USA (2015)
4. Yifan Hu, Shi Lei, Visualizing Large Graphs. Wiley Interdisciplinary Reviews: Computational Statistics, Volume 7, pages 115–136 (2015)
5. Charles D. Stolper, Minsuk Kahng, Zhiyuan Lin, Florian Foerster, Aakash Goel, John Stasko, Duen Horng Chau, GLOSTIX: Graph-Level Operations for Specifying Techniques and Interactive eXploration. Transactions on Visualization and Computer Graphics, IEEE Computer Society, Los Alamitos, CA, USA (2014)
6. Michael McGuffin and Igor Jurisica, Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations. IEEE Transactions on Visualization and Computer Graphics, pages 937–944 (2009)
7. Tim Dwyer, Bongshin Lee, Danyel Fisher, Kori Inkpen Quinn, Petra Isenberg, George G. Robertson, Chris North, A Comparison of User-Generated and Automatic Graph Layouts. IEEE Transactions on Visualization and Computer Graphics, pages 961–968 (2009)
8. Wendy T. Lucas and Taylor Gordon, User Control of Force-directed Layouts. Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016), Volume 1: ICSOFTEA, Lisbon, Portugal (2016)
9. Arlind Nocaj, Mark Ortmann, Ulrik Brandes, Untangling the Hairballs of Multi-Centered, Small-World Online Social Media Networks. Journal of Graph Algorithms and Applications (2015)
10. Quang Vinh Nguyen, Mao Lin Huang, A New Interactive Platform for Visual Analytics of Social Networks. Proceedings of the Visual Information Communications International 2009 (VINCI09), Sydney, Australia (2009)

11. Leandro Moraes, V. Cruz, Luiz Velho, A Sketch on Sketch-Based Interfaces and Modeling. 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials, pages 22–33, IEEE Computer Society, Los Alamitos, CA, USA (2010)
12. Sebastian Schmidt, Miguel A. Nacenta, Raimund Dachselt, Sheelagh Carpendale, A Set of Multitouch Graph Interaction Techniques. International Conference on Interactive Tabletops and Surfaces (ITS10), New York, NY, USA (2010)
13. Bertin J., Semiologie graphique: les diagrammes, les reseaux, les cartes. Mouton, Paris, France (1967)
14. Burch M., Fritz M., Beck F., and Diehl S., Timespidertrees: A novel visual metaphor for dynamic compound graphs. In Visual Languages and Human-Centric Computing (VL/HCC), IEEE Symposium, pages 168–175 (2010)
15. Burch M., Beck F., and Weiskopf D., Radial edge splatting for visualizing dynamic directed graphs. In GRAPP/IVAPP, pages 603–612 (2012)
16. Vehlow C., Burch M., Schmauder H., and Weiskopf D., Radial layered matrix visualization of dynamic graphs. In Information Visualisation (IV), 17th International Conference, pages 51–58, IEEE (2013)
17. Holten, Danny, Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. In IEEE Trans. on Visualization and Computer Graphics, pages 741–748, IEEE Educational Activities Department, Piscataway, NJ, USA (2006)
18. Berry G., Real Time Programming: Special Purpose or General Purpose Languages (1989)
19. Tufte, E., Envisioning Information. Graphics Press, Cheshire, CT, USA (1990)
20. Zhao J., Chevalier F., and Balakrishnan R., Kronominer: using multi-foci navigation for the visual exploration of time-series data. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1737–1746, ACM (2011)
21. Diehl S., Beck F., and Burch M., Uncovering strengths and weaknesses of radial visualizations: an empirical approach. IEEE Transactions on Visualization and Computer Graphics, 16(6):935–942 (2010)
22. Dragicevic P. and Huot S., Spiraclock: a continuous and non-intrusive display for upcoming events. In CHI'02 extended abstracts on Human factors in computing systems, pages 604–605, ACM (2002)
23. Lin, Y. and Vuillemot, R., Spirograph designs for ambient display of tweets. In IEEE VIS 2013, Atlanta, GA, United States, IEEE (2013)
24. Krzywinski M., Schein J., Birol I., Connors J., Gascoyne R., Horsman D., Jones S. J., and Marra M. A., Circos: an information aesthetic for comparative genomics. Genome research, 19(9):1639–1645 (2009)
25. Krzywinski M., Birol I., Jones S. J., and Marra M. A. Hive plots: rational approach to visualizing networks. Briefings in Bioinformatics, 13(5):627 (2012)
26. Beck F., Burch M., Diehl S., and Weiskopf D., A taxonomy and survey of dynamic graph visualization. Computer Graphics Forum, 36(1):133–159 (2017)
27. Wu Y., Wei F., Liu S., Au N., Cui W., Zhou H., and Qu H., Opinionseer: interactive visualization of hotel customer feedback. IEEE transactions on visualization and computer graphics, 16(6):1109–1118 (2010)
28. Klamt S., Haus U.-U., and Theis F., Hypergraphs and cellular networks. PLoS Comput Biol, 5(5) (2009)
29. Saket B., Scheidegger C., and Kobourov S. G., Comparing node-link and node-link-group visualizations from an enjoyment perspective. Comput. Graph. Forum, 35(3):41–50 (2016)
30. Fruchterman T. M. J. and Reingold E. M., Graph drawing by force-directed placement. Softw. Pract. Exper., 21(11):1129–1164 (1991)