Physical Navigation to Support Graph Exploration on a Large High-Resolution Display

Anke Lehmann, Heidrun Schumann, Oliver Staadt, Christian Tominski

Institute for Computer Science
University of Rostock
A.-Einstein-Str. 21, 18059 Rostock, Germany
http://vcg.informatik.uni-rostock.de

Abstract. Large high-resolution displays are potentially useful to present complex data at a higher level of detail embedded in the context of surrounding information. This requires an appropriate visualization and also suitable interaction techniques.
In this paper, we describe an approach to visualize a graph hierarchy on a large high-resolution display and to interact with the visualization by physical navigation. The visualization is based on a node-link diagram with dynamically computed labels. We utilize head tracking to allow users to explore the graph hierarchy at different levels of abstraction. Detailed information is displayed when the user is closer to the display and aggregate views of higher levels of abstraction are obtained by stepping back. The head tracking information is also utilized for steering the dynamic labeling depending on the user's position and orientation.

Key words: Information Visualization, Interaction, Graph Exploration, Large High-Resolution Displays, Head Tracking

1 Introduction

Visualization is challenged by the fact that the available display space usually cannot keep up with the amount of information to be displayed. Commonly this problem is addressed by a kind of overview+detail technique [1], where users interactively switch between an abstract overview of the entire data and detailed views of smaller parts of the data. Large high-resolution displays (LHRDs) are an emerging technology and a promising alternative to the classic overview+detail approaches. LHRDs combine a large physical display area with high pixel density for data visualization (see Ni et al. [2]). A unique feature of LHRDs is that they allow users to perceive the global context of complex information presented on the display by stepping back, while enabling them to explore finer detailed data by stepping closer. This makes LHRDs suitable for visualization applications, such as the visual exploration of graphs.

However, because classic mouse and keyboard interaction is infeasible in LHRD environments, we require new approaches to interact with the visualization. Ball et al. [3] found that physical navigation is quite useful in this regard.

We utilize this fact for a basic exploration task. In particular, the user's physical position in front of a LHRD determines the level of abstraction of the visual representation of hierarchical graphs. Graph nodes are expanded dynamically when the user moves closer to the display (i.e., more detail). When the user steps back, nodes are collapsed, by which a higher level of abstraction (i.e., less detail) is obtained. This provides a natural way to get an overview or detailed information similar to virtual zoom interaction via mouse's scroll wheel. A benefit of our approach is that the users do not need their hands for the interaction, which opens up possibilities to use them for other tasks.

Switching between different levels of abstraction also requires adaptation of the visual representation. While the operations expand and collapse correspond to addition and removal of nodes in the graph layout, respectively, the question remains how to appropriately label the differently abstracted objects shown on the LHRD. We utilize an existing labeling algorithm (i) to ensure readability of labels depending on the level of abstraction (determined by the user display distance) and (ii) to avoid readability problems caused by LCD-panel-based LHRDs.

In the next section, we will briefly describe basics of graph exploration and related work on interactive visualization on LHRDs. In Section 3, we will describe details of our approach to support graph exploration on LHRDs. Section 4 will outline the design of our prototype system and preliminary user feedback. We will conclude with a summary and an outlook on future extensions and applications of our approach in Section 5.

2 Related Work

Our literature review is structured as follows. We briefly describe graph exploration in general and then consider visualization on large displays and shed some light on what modern input concepts beyond mouse and keyboard input can offer for interactive visualization.

2.1 Graph Exploration

Lee et al. [4] identified several tasks that users seek to accomplish with the help of graph visualization. Lee et al. structure low-level tasks by means of a taxonomy with three high-level categories: topology-based tasks, attribute-based tasks, and browsing-tasks. In each of these categories, we find tasks of exploratory nature, such as "find adjacent nodes", "find edge with largest weight", "follow a given path", or "return to previously visited node".

In general, interactive graph exploration is supported by appropriate visualization techniques (see Herman et al. [5]) and well-known overview+detail and focus+context interaction (see Cockburn et al. [1]). Such techniques enable users to zoom in order to view details or an overview, and to pan in order to visit different parts of the data.

A common approach to support the exploration of larger graphs is to structure (e.g., to cluster or aggregate) them hierarchically (see Herman et al. [5]). Elmqvist and Fekete [6] describe the various advantages of this approach, including the fact that it allows us to visualize different abstractions of the underlying data. In order to access the level of abstraction that is required for the task at hand, users interactively expand or collapse individual nodes of the hierarchy or switch between entire levels of the hierarchy.

Additionally, special lens techniques have been proposed for interacting in locally restricted areas of the visualization. For instance, van Ham and van Wijk [7] use a virtual lens to automatically adjust the level of abstraction (i.e., expand/collapse nodes) depending on the position of the lens. Tominski et al. [8] and Moscovich et al. [9] describe lenses that create local overviews of the neighborhood of selected nodes and provide direct navigation along paths through a graph.

All of these examples demonstrate the usefulness of dedicated interaction techniques for graph exploration. However, most of the existing techniques have been designed for classic desktop environments with regular displays and mouse interaction. In this work, we adapt existing concepts to make them applicable in LHRD environments. The exploration in terms of different levels of abstraction is of particular interest to us, because of the inherent overview+detail capabilities of LHRDs.

2.2 Interactive Visualization on Large Displays

The advantages of LHRDs (i.e., many pixels and natural overview+detail) make them an interesting alternative to desktop-based visualization scenarios. Next, we review existing approaches that utilize LHRDs for visualization.

As demonstrated by Keim et al. [10], pixel-based visualization approaches are a good example of visualization techniques that benefit from the larger number of pixels available on LHRDs. Another example is the visualization of complex spatiotemporal data. Booker et al. [11] explain that the exploration of spatiotemporal data can significantly benefit from LHRDs. They also conjecture major benefits for other information exploration scenarios. While these examples focus on the output capabilities of LHRDs, other researchers have addressed questions of interaction.

While mostly hand-held devices (e.g., wireless mouse, tracked button device) are used to pan or zoom in a desktop-based visualization application, such devices, if at all, are cumbersome to use in LHRD environments. A commonly accepted means to drive interaction in LHRD is tracking the user's physical movements. Vogel and Balakrishnan [12] use head tracking to switch between the display of ambient, public, or personal information depending on the user's distance in front of a large display. Ashdown [13] switch the mouse pointer between monitors by head tracking to increase mouse movements in multi-monitor environments. Ball et al. [14, 3] identify that in LHRDs the performance of simple navigation tasks for finely detailed data can be increased by using physical navigation. This knowledge is realized by Peck et al. [15] who adjust the mechanisms for interactive selection and navigation based on the user's distance to the display. A survey of interaction techniques for LHRDs is presented by Khan [16].

Although physical interaction has a number of advantages, there are some limitations. For example, Ball et al. [17] use head rotation as input for panning navigation in a geospatial visualization application. However, virtual navigation and head motion were tightly coupled, which made it difficult for users to pan the map while also scanning it. Consequently, head tracking is better suited for simple interaction mechanisms than for fine motor control.

Overall, prior work shows that using physical navigation in LHRD environments can improve both perception of the visualized data [14, 3] and interaction with the visualization [15]. Considering these advantages, we devised an interactive visualization approach to support the exploration of graphs on LHRDs at different levels of abstraction.

3 Exploring Graphs on a Large High-Resolution Display

In recent years, graphs have gained importance in many application backgrounds such as social networks, power networks, climate networks, biological networks and others. We present an approach that is suitable to support graph exploration by exploiting the advantages of LHRDs and physical navigation.

First we will explain the data that we address, which are basically graph hierarchies. Then we describe the node-link-based visualization employed in this work. Finally, we introduce novel interaction techniques for adjusting the level of abstraction of the displayed data based on head tracking.

3.1 Data

We use a *graph hierarchy* H as the main data structure to drive the exploration [5, 6]. H is a rooted tree whose leaves represent information on the finest level of granularity. Non-leaves of H are abstractions of their corresponding child nodes. Any "full cut" through H defines a view of the data with a specific level of abstraction.

There are two basic means to enable users to choose a suitable level of abstraction: (i) one can globally switch from one level of the hierarchy to another or (ii) one can expand and collapse nodes in order to adjust the level of abstraction locally. Switching the level globally means replacing all nodes of the current level with the nodes of another level. On the other hand, expansion of a non-leave node locally replaces that node with its children, which results in more information and less abstraction. Collapsing a set of nodes replaces these nodes with their parent node, which results in less information and more abstraction.

For the purpose of demonstration, we visualize the hierarchy of the ACM computing classification system, where nodes correspond to text labels of the categories and edges between nodes indicate related categories[1]. With its 1473 nodes and 464 edges this data set is not too large and the explicitly given labels on the different levels of the hierarchy are expressive and easy to understand. Both facts make this data set quite useful for first experiments.

Note that edges have been added by hand, which is the reason why an edge that exists on a higher level of abstraction may have no corresponding edges on lower levels. In particular, no edges exist on the finest granularity of the ACM data set.

3.2 Visualization

The visualization relies on a basic node-link representation (see Figure 1). The required graph layout has been pre-computed using a recursive hybrid algorithm: A force-directed mechanism determined node positions for connected parts of the graph [18] and a variant of the squarified treemap layout handled unconnected parts [19]. In a second phase, node positions were adjusted manually to better echo the semantics of the data (e.g., the ordering of categories). Third, an automatic local adjustment of the layout is computed to avoid that nodes are placed behind bezels of our LCD-panel-based LHRD.

In our node-link visualization, nodes are visualized as spheres. To separate the various hierarchy levels, the spheres are colored depending on the depth of the node in the graph hierarchy using a sequential color scheme from ColorBrewer[2]. The links between spheres are shown as colored lines. Along a line, we interpolate the color of its two incident spheres. This way, edges between nodes of the same level in the hierarchy (i.e., no interpolation, because both spheres have the same color) are clearly distinguishable from edges between nodes of different levels

[1] http://dspace-dev.dsi.uminho.pt:8080/en/addon_acmccs98.jsp
[2] http://www.colorbrewer.org

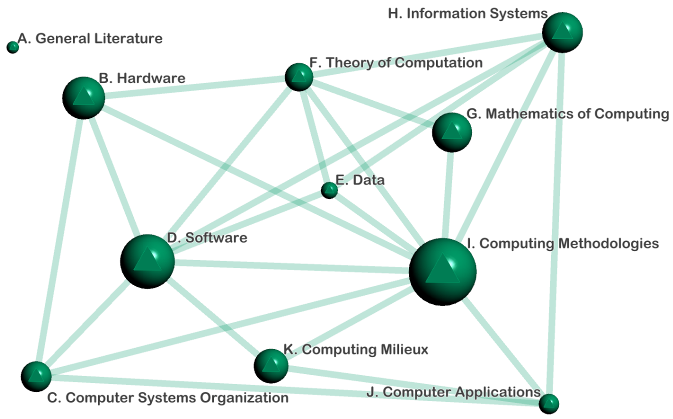

Fig. 1. Node-link view of the top level of the ACM computing classification system.

(i.e. color interpolation, because sphere colors are different). Although the layout already communicates the hierarchical structure of the data quite well, we use convex hulls as visual envelopes for nodes that share the same parent.

Because we work with a graph whose nodes represent category captions, we have to attend to label placement. This is also relevant when it comes to showing text labels with associated node properties (e.g., node degree, node depth, etc.). To ensure label readability, we have to account for two aspects. First, we have to deal with bezels of LCD-panel-based LHRDs. Usually, bezels are handled as part of the display space and an empty virtual space is placed behind them in order to make virtual objects (e.g., spheres) drawn across monitors appear more natural and undistorted. What might be good for virtual objects is unfavorable for labeling, because we risk loosing important textual information in the empty virtual space behind the bezels. In order to avoid this, labels must not overlap the bezels. Secondly, we have to account for the larger range of distances from which labels must be readable. This requires adjustment of label sizes depending on the viewing distance.

Due these requirements we need a dynamic calculation of the labeling at real-time. We employ the labeling algorithm by Luboschik et al. [20], which satisfies our needs. The algorithm allows the placement of so-called conflict particles in those parts of the labeling space where labels must not appear. We insert such particles exactly where the bezels are located. Hence, we can guarantee that labels never overlap with bezels. However, labels that are larger than the size of a single LCD-panel cannot be placed anymore. To mitigate this problem, we split labels where possible. Once the label positions have been calculated, we use scalable vector fonts for the text rendering.

Given the number of pixels of LHRDs, we are theoretically able to visualize the entire data. However, in that case, it might be difficult for viewers to recognize the multitude of graph elements. Moreover, as in our example, interesting findings might be revealed on different levels of a graph hierarchy. Therefore, it is important to enable the user to explore the data interactively.

3.3 Interaction

When visually exploring a graph hierarchy, the adjustment of the level of abstraction is an essential interaction. We realize this interaction based on head tracking. By using head tracking we obtain information about the user's head position and orientation (6 degrees of freedom) in front of the display wall. As mentioned in Section 2.2, care must be taken that the interaction be robust against small head motions. We implemented two alternative methods to utilize the head tracking information: (1) the zone technique and (2) the lens technique.

Both techniques allow the user to change the level of abstraction of the displayed graph hierarchy by moving in front of the LHRD. Transitions from one level to another are animated to retain the user's mental map. The zone technique corresponds to a level-wise global adjustment, whereas the lens technique realizes a local adjustment in those parts of the visualization that the user is cur-

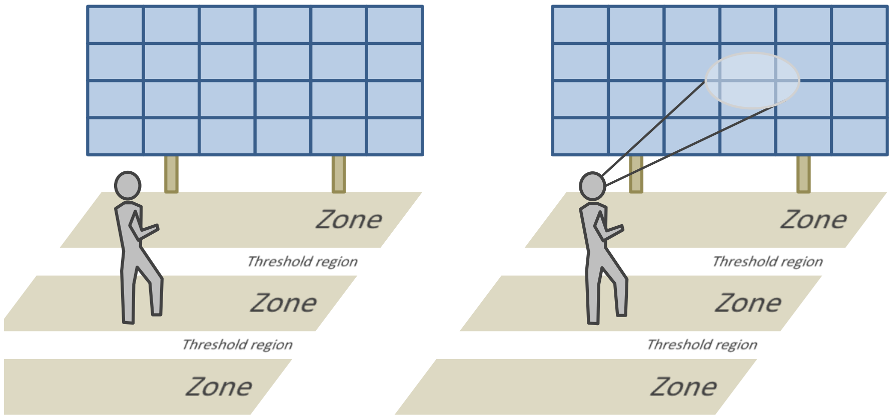

Fig. 2. Illustration of zone technique (left) and lens technique (right).

rently looking at. This input offers a hands-free interaction method, where the hands can be used for finer motor control tasks (e.g., interactive manipulation).

The zone technique supports intuitive level switching based on head tracking, where only the user's distance is considered. For the zone technique, the interaction space in front of the wall is divided into parallel zones, one zone for each level. The interaction zone for the root of the graph hierarchy is farthest away from the display, the zone for the deepest leaves of the graph hierarchy is closest to the display. As the user navigates physically (i.e., step forward or backward) we switch the level of the graph hierarchy. The closer the user moves toward the display the more details are displayed. In order to avoid sudden unintentional switches, additional threshold regions are inserted between two adjacent zones (see Figure 2, left). Thus the user has to cross the threshold region entirely to release a switching.

The advantage of the zone technique is a natural changing of the level of detail by forward and backward moves. However, the user changes the level of detail globally for the entire graph. This reveals edges within the same level, but edges across different levels remain hidden (e.g., a relation between a node and the children of another node).

The lens technique facilitates to keep the overall context, while providing detailed information for a user-selected part of the data. The selection is done by an approximation of the gaze direction. Head tracking delivers information about the head position and orientation. Consequently, we create a viewing cone from the user to the display screen to approximate the user's field of view (see Figure 2, right). As a visual feedback, the field of view is shown as a lens ellipse on the LHRD (see Figure 3).

Fig. 3. Exploring a graph with the lens technique.

As with the zone technique, the distance determines the currently displayed level of abstraction. By steering the lens with small head movements, the user is able to scan the graph and get insight into detailed information in different parts of the data. Nodes that enter the lens are dynamically expanded to reveal the next lower level of the graph hierarchy (i.e., more detail). When a node exits the lens it is collapsed to return to the original level of detail.

To keep the amount of displayed information understandable, the lens size is adjusted to the user's distance from the LHRD. When the user steps closer to the display, the lens size decreases. This naturally matches with the smaller field of view that the user covers when standing close to the LHRD.

The lens technique offers an interesting focus+context interaction, where the overall context is preserved while the user accesses details by moving the head. However, unintentional head movements can effect frequent expand/collapse operations, which cause a kind of disturbing flicker. Therefore, we enabled the Kalman filter provided by the tracking software to reduce the natural head tremor.

Both of our interaction techniques require that the labels be adjusted depending on the user distance (and level of abstraction) in order to improve readability. When the user is close to the display we can decrease the font size to free display space, which allows the algorithm to place more labels (see Figure 3). In contrast, when the user is far away from the display, small labels are unreadable. In such a setting, we traverse up in the graph hierarchy and pick aggregated labels, which are rendered using a larger font face. We use the common "Arial" font in a range from 12pt to 32pt.

4 Prototype and Preliminary User Feedback

We implemented a prototype to study the visual exploration technique on our LHRD. Next we describe the technical details and report on preliminary user feedback.

4.1 Technical Details

We use a flat tiled LCD wall consisting of 24 DELL 2709W displays, where each tile has a resolution of 1920 × 1200 pixels, with a total resolution of 11520 × 4800 (55 million pixels). The interaction space in front of the display wall (see Figure 4) has an area of approximately 3.7m × 2.0m and a height of about 3.5m. The displays are connected to a cluster of six render nodes (slaves) and one additional head node (master). For the rendering, we utilize the graphics framework CGLX[3] and the font rendering library FTGL[4].

Our prototype uses an infrared tracking system (Naturalpoint OptiTrack) with 12 cameras, which are arranged semicircularly around the interaction space. The user wears a baseball cap that has attached to it reflective markers. Tracking these markers enables us to determine the position and the orientation of the user's head in the interaction space.

4.2 User Feedback

Using the aforementioned prototype, we collected preliminary user feedback. We asked computer science students (one female and seven male students) to test the application with the described interaction techniques about ten minutes. The users explored the graph (see Section 3.1) by walking around in the interaction space. They could switch between the zone technique and the lens technique by flipping the baseball cap during the trial. In the interview afterwards the participants were asked about readability of labels from any distance, ease-of-use, and their preferred technique.

All participants indicated that the head tracking was easy to use and that the labels were readable from any distance. The participants reported that the zone technique was easier to use because there were no unintentional level switches during the interaction. This indicates to us that the threshold regions for the zone technique are effective. Although the zone technique was easier to control, the subjects found it less interactive and they felt that there was too much information in their peripheral visual field.

The lens was named as an eye-catcher and as being more interactive than the zone technique. Providing detail information inside the lens while maintaing the context outside the lens helped users to cope with the amount of information. On the other hand, the lens technique required a head calibration for every user to enable a reliable interaction behavior. Moreover, even after careful calibration,

[3] http://vis.ucsd.edu/∼cglx/
[4] http://sourceforge.net/projects/ftgl/

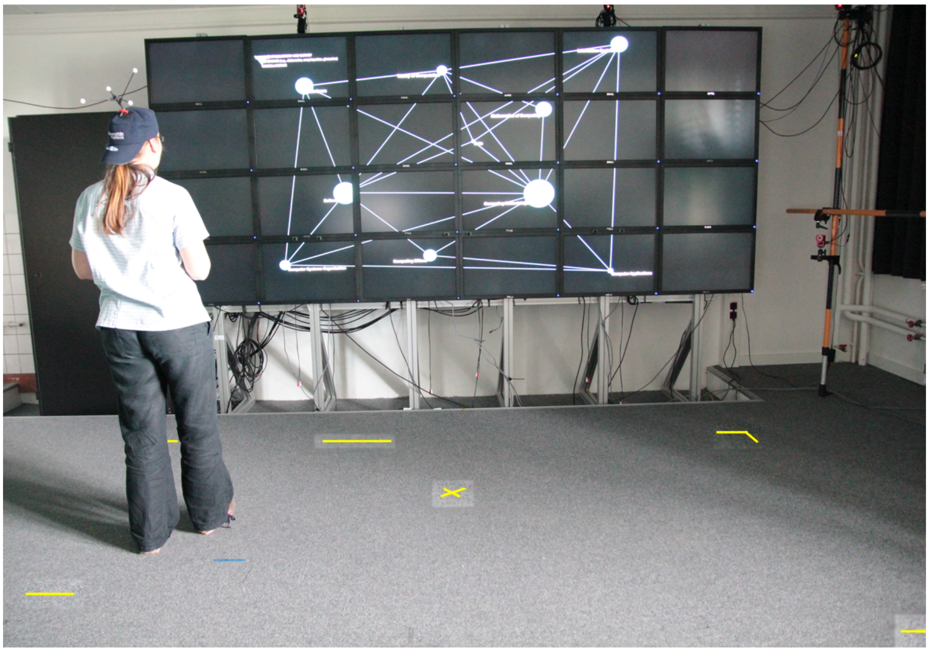

Fig. 4. Tiled LCD wall and interaction space indicated with floor marks.

the users experienced unintended interaction caused by small head motions. This feedback suggests to us that we require improved filter methods to mitigate the effects of natural head tremor.

5 Summary and Future Work

We have introduced an approach to support the exploration of graphs on large high-resolution displays. The visualization is dynamically adjusted as the user moves in front of the display. Based on head tracking input, the labeling of the visual representation is recomputed to ensure label readability. Moreover, we implemented two interaction techniques based on physical navigation. Both the zone technique and the lens technique utilize the user's position to determine the displayed level of abstraction. Additionally, the lens technique considers an approximation of the viewing direction to provide more details for a selected part of the data while preserving the overall context. Preliminary user feedback indicates that users can easily learn and apply the developed techniques. Our prototype is capable of showing two thousand nodes and labels at interactive frame rates.

We understand our work as an initial step to be followed by further research on visualization on LHRDs. The next step is to integrate a very large data set (e.g., a metabolic pathway from systems biology) into the LHRD environment and to perform a detailed user study.

There are a number of general questions to be answered as well as specific issues to be addressed. In terms of the visualization, we have to improve our algorithmic solutions for avoiding overlap of important visual information with bezels. We already achieved quite good results for the labeling and are confident that similar solutions can be found for clusters of nodes and edges of graphs. Further investigations are necessary to find general solutions for other visualization techniques. This also has to include the enhancement of existing methods to better utilize the available pixels.

The large display space and the interaction space in front of the display are potentially useful for collaborative work. We have experimented with a setting where two persons collaborate. In such a setting, we have to consider enhanced visual feedback of the participating users and conflicts in the determination of the level of detail to be shown – globally as well as locally. However, this work is still in a very early stage. In future projects, we will investigate ways to enable users to explore graphs with physical navigation in a multi-user scenario.

More specifically, with regard to our prototype, we plan to consider further interaction tasks to be accomplished in the LHRD environment. For instance, the user should be able to freeze selected nodes for the purpose of visual comparison. We are also thinking of manipulation techniques for graphs to allow users to edit the data. It is also conceivable that visualization parameters (e.g., font size) are interactively customizable within limits. We conjecture that the so far unused hands of the user are most suitable for these finer interaction tasks.

Acknowledgments. This work was supported by EFRE fond of the European Community and a grant of the German National Research Foundation (DFG), Graduate School 1424 MuSAMA. We gratefully acknowledge implementation contributions by Thomas Gertz. We are also grateful for the valuable comments by the anonymous reviewers.

References

1. Cockburn, A., Karlson, A., Bederson, B.B.: A review of overview+detail, zooming, and focus+context interfaces. ACM Computing Surveys **41** (2009) 2:1–2:31
2. Ni, T., Schmidt, G., Staadt, O., Livingston, M., Ball, R., May, R.: A survey of large high-resolution display technologies, techniques, and applications. In: Virtual Reality Conference, 2006. (2006) 223 – 236
3. Ball, R., North, C., Bowman, D.A.: Move to improve: promoting physical navigation to increase user performance with large displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2007)
4. Lee, B., Plaisant, C., Parr, C.S., Fekete, J.D., Henry, N.: Task taxonomy for graph visualization. In: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization. BELIV '06, New York, NY, USA, ACM (2006) 1–5

5. Herman, I., Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics **6** (2000)
6. Elmqvist, N., Fekete, J.D.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Transactions on Visualization and Computer Graphics **16** (2010)
7. van Ham, F., van Wijk, J.J.: Interactive visualization of small world graphs. In: Proceedings of the IEEE Symposium on Information Visualization (InfoVis), IEEE Computer Society (2004) 199–206
8. Tominski, C., Abello, J., Schumann, H.: Cgv – an interactive graph visualization system. Computers & Graphics **33** (2009)
9. Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., Fekete, J.: Topology-aware navigation in large networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), ACM (2009)
10. Keim, D.A., Schneidewind, J., Sips, M.: Scalable pixel based visual data exploration. In: Pixelization Paradigm. Volume 4370 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2007) 12–24
11. Booker, J., Buennemeyer, T., Sabri, A.J., North, C.: High-resolution displays enhancing geo-temporal data visualizations. In: Proceedings of the ACM Southeast Regional Conference. (2007) 443–448
12. Vogel, D., Balakrishnan, R.: Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: Proceedings of the 17th annual ACM symposium on User interface software and technology. UIST '04, New York, NY, USA, ACM (2004) 137–146
13. Ashdown, M., Oka, K., Sato, Y.: Combining head tracking and mouse input for a gui on multiple monitors. In: CHI '05 extended abstracts on Human factors in computing systems. CHI EA '05, New York, NY, USA, ACM (2005) 1188–1191
14. Ball, R., North, C.: Effects of tiled high-resolution display on basic visualization and navigation tasks. In: CHI '05 extended abstracts on Human factors in computing systems. CHI '05, New York, NY, USA, ACM (2005) 1196–1199
15. Peck, S.M., North, C., Bowman, D.: A multiscale interaction technique for large, high-resolution displays. In: Proceedings of the 2009 IEEE Symposium on 3D User Interfaces. 3DUI '09, Washington, DC, USA, IEEE Computer Society (2009) 31–38
16. Khan, T.K.: A survey of interaction techniques and devices for large high resolution displays. In Middel, A., Scheler, I., Hagen, H., eds.: Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop). Volume 19 of OpenAccess Series in Informatics (OASIcs)., Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2011) 27–35
17. Ball, R., Dellanoce, M., Ni, T., Quek, F., North, C.: Applying embodied interaction and usability engineering to visualization on large displays. In: ACM British HCI - Workshop on Visualization & Interaction. (2006) 57–65
18. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall (1999)
19. Bruls, M., Huizing, K., van Wijk, J.J.: Squarified treemaps. In: Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym), Eurographics Association (2000)
20. Luboschik, M., Schumann, H., Coords, H.: Particle-based labeling: Fast point-feature labeling without obscuring other visual features. IEEE Transactions on Visualization and Computer Graphics **14** (2008)