Multi-Touch Manipulation of Magic Lenses for Information Visualization

Ulrike Kister
Interactive Media Lab
Technische Universität Dresden
Dresden, Germany
ukister@acm.org

Patrick Reipschläger
Interactive Media Lab
Technische Universität Dresden
Dresden, Germany
patrick.reipschlaeger@
tu-dresden.de

Raimund Dachselt
Interactive Media Lab
Technische Universität Dresden
Dresden, Germany
dachselt@acm.org

Figure 1: Multiple lenses can be applied to explore the data set.

Abstract

We introduce touch-enabled magic lenses that can be manipulated and parametrized through fluent interactions. Interaction with lenses for information visualization and data exploration has mostly been limited to single-user, single-function lenses. In this work, we present our prototype on lenses where lens function, parameters and combination of functions can be manipulated using fluent touch interaction. To achieve this, our tool consists of a widget-based approach for novice users as well as a set of continuous gestures for expert users. Additionally, we support the combination of lenses and thereby create a multi-purpose lens tool.

Author Keywords

Magic lenses; graph lenses; multi-touch interaction; information visualization.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces.

Introduction

Magic lenses [1] are focus and context tools that change the visual representation of the data in a region of interest within the visualization. Recently, Tominski et al. [8] presented a state-of-the-art report on interactive lenses in

visualization showing a multitude of application cases and data types where lenses can be helpful tools for exploration.

In previous work, lenses have mostly been discussed in terms of their individual lens filter. Few researchers have explored interaction with lenses, with mouse interaction being the prevalent technique for placing and adjusting magic lenses. However, as interactive surfaces become more common, we focus on using the potential of these devices for interactive lens manipulations.

This is not the first paper addressing lenses on interactive surfaces. However, they have mainly been applied to improve touch selection by magnifying a region of interest. One example of this is the FingerGlass approach by Käser et al. [3] who address target selection by presenting fluent bimanual interactions for lens creation and repositioning. For information visualization, lens manipulation has been applied recently to scatterplot exploration on tablets [5]. Here, a magnification lens is created and the magnification factor is set by dragging an element at the border of the lens.

Various lens filters have been proposed in research (e.g. [1, 6, 7]), magnification being only one of them. Often lens filters are specifically designed to give insight into a selected data type [8]. Each lens filter can have different filter-dependent parameters whose manipulation enhances the outcome of the filter. Additionally, the combination of different lens filters can yield better exploration tools [7]. To our knowledge, no work exists that discusses interaction for lens parameter adjustment and lens combination.

With this work, we contribute 1) fluent interaction with lenses, focusing on both novice and expert users,

2) touch-based techniques for lens adjustment, where we focus on the lens filter as a changeable attribute that can be switched and adjusted in place on the lens, and 3) the combination of different lens filters to form one coherent lens exploration tool.

Touch-enabled Magic Lenses

For the design of our lenses, we concentrated on creating consistent gestures and support fluent interaction for both novice and expert users alike. We developed each lens as an individual tool without global menus to support multiple active lenses and multi-user scenarios (Figure 1).

The interactive lenses presented in this work are circular or elliptical shaped tools. They are visually represented by a very thin visual border as to not occlude the data presented underneath. The border color encodes the currently selected lens filters (see Figure 3). The border is transparently enlarged to allow basic touch manipulations, such as drag for repositioning, pinch for scaling and two finger rotation.
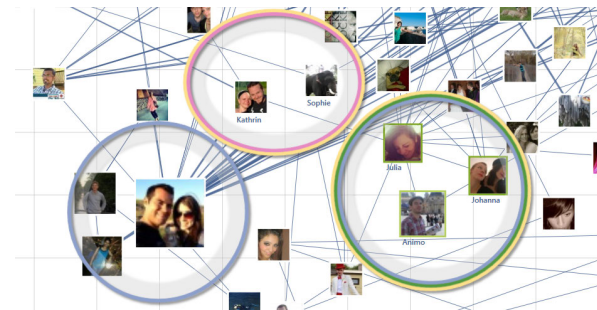

Figure 2: a) The radial touch menu presents different lens filters and their parameters. b) Parameters can be adjusted using a slider which always snaps back to its original position.

Figure 3: Lenses are shapes with a thin colored border. The color of the border describes the currently active lens filter. Lenses can have one single or multiple lens filters.
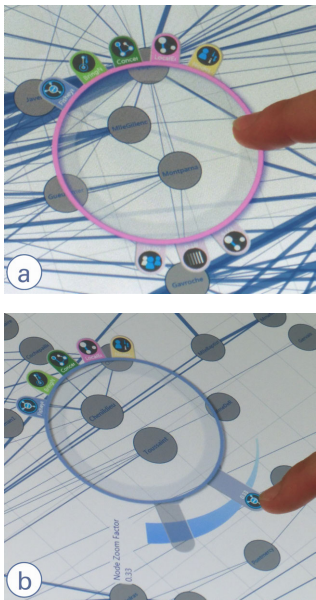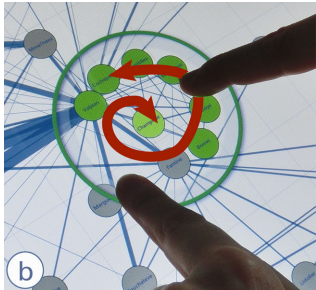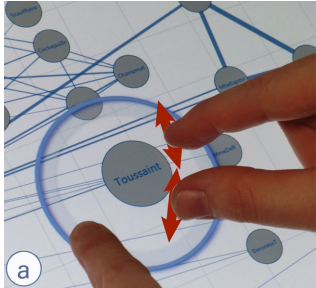
Figure 4: Two example gestures (symbolized by red arrows) who trigger a new lens function and adjust the main parameter.
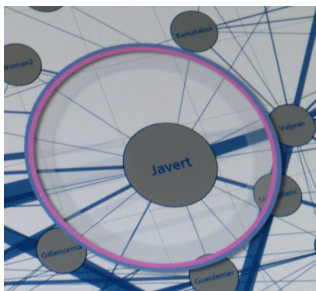

Figure 5: Lenses can be dragged on top of each other to combine their filters and create one unified lens.

Radial Touch Menu
We developed an approach of externalizing the lens filters and their parameters to the border of the lens. This way all parameters of the lens are visible to novice users. The radial touch menu is separated into a top part and a section at the bottom of the lens (see Figure 2a). In the upper menu, lens filters can be selected using a tap or drag the handle out from the lens. Different colors and icons support the recognition of the available lens filters.

In the bottom menu the parameters of the active filters are represented by their individual icon and a less saturated version of their filter color. These handles can be dragged out further and transform into be a slider thumb. They can be rotated around the border of the lens to adjust the individual filter parameter (see Figure 2b). When lifting the finger, the handle automatically snaps back to its original position. On renewed pull-out the slider has automatically adjusted, so that relative positioning is possible again.

Continuous Gestures
For efficient expert use, we also designed a gesture set where each gesture triggers a new lens filter. We focused on creating efficient and fluent interactions that can be grouped together to form a phrase [2]. Hence, we selected continuous gestures whose continuation immediately adjusts the main parameter of the activated filter. We present only two of the possible gestures in this work, to illustrate this principle. To activate a Fisheye magnification, we selected a simple pinch-gesture (see Figure 4a) where the relative distance between the two touch points adjusts the amount of distortion. A Bring Neighbors [7] filter can be triggered by dragging in a spiral movement (see Figure 4b), similar to the gesture in CycloZoom+ [4]. It symbolizes the pulling in of adjacent

nodes, neighbors, for graph nodes. How much neighbors are drawn in is adjusted by the distance of the touch point to the center of the lens.

Combining Lenses
Due to insight from user tests, Tominski et al. [7] presented a *Composite Lens* which incorporates three fixed lens filters into one lens. We designed our touch-enabled magic lenses to allow flexible combination of lens filters. Lenses of about the same size can be dragged on top of each other and then snap together to form one multi-filter lens. Additionally, it is possible to simply add a filter to an existing lens using the previously described gesture set. We use a second finger on the border of the lens to distinguish the addition of filters from the simple replacement of the current lens filter. Figure 5 shows the combination of Fisheye [6] and Local Edge [7] filter for graph exploration, enlarging the node in focus and removing edge clutter simultaneously within one lens.

Implementation
To enable users to explore graph data and test our fluent interaction principle, we implemented our touch-enabled magic lens concept as a C#/WPF application running on a 27" Perceptive Pixel[1] display. We used graph exploration as an application case and implemented the graph visualization using QuickGraph[2] as graph toolkit and Graph#[3] for layouting purposes. Possible data sets are basic example graphs in graphML format, e.g., showing co-occurring diseases, or social networks in JSON format generated by the Facebook Graph API Explorer[4].

---

[1]http://www.perceptivepixel.com/
[2]http://quickgraph.codeplex.com/
[3]http://graphsharp.codeplex.com/
[4]http://developers.facebook.com/tools/explorer

Both the radial touch menu as well as the continuous gestures are implemented with currently five filters related to graph exploration. The radial menu and the continuous gesture set form one coherent concept where switching between gestures and menu-based manipulation is possible at any moment of interaction.

Conclusion

We present a graph exploration prototype demonstrating our concepts of fluent interaction with magic lenses. Unlike previous work, we focus on adjustment of the lens filter and its filter-dependent parameters. We present a widget-based approach that externalizes all possible filters and parameters to the border of the lens. This specifically supports novice users as all parameters are visible. We consciously refrain from using global menus to account for multi-user scenarios and large display setups. Additionally, we focused on a set of continuous gestures for expert users. These gestures allow activation of other filters, either by adding to or replacing the current filters. Allowing continuation after recognition, these gestures also adjust the main parameter of each lens filter. While our prototype is geared to graph exploration, the principles show great potential to be applied to other application areas.

Acknowledgements

References

[1] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *Proc.*, SIGGRAPH '93, ACM (1993), 73–80.

[2] Buxton, W. Chunking and phrasing and the design of human-computer dialogues. In *Proc. IFIP World Computer Congress*, North Holland Publishers (1986), 475–480.

[3] Käser, D. P., Agrawala, M., and Pauly, M. Fingerglass: efficient multiscale interaction on multitouch screens. In *Proc.*, CHI '11, ACM (2011), 1601–1610.

[4] Malacria, S., Lecolinet, E., and Guiard, Y. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. In *Proc.*, CHI '10, ACM (2010), 2615–2624.

[5] Sadana, R., and Stasko, J. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proc.*, AVI '14, ACM (2014), 265–272.

[6] Sarkar, M., and Brown, M. H. Graphical fisheye views of graphs. In *Proc.*, CHI '92, ACM (1992), 83–91.

[7] Tominski, C., Abello, J., van Ham, F., and Schumann, H. Fisheye tree views and lenses for graph visualization. In *Proc.*, IV '06, IEEE Computer Society (Washington, DC, USA, 2006), 17–24.

[8] Tominski, C., Gladisch, S., Kister, U., Dachselt, R., and Schumann, H. A survey on interactive lenses in visualization. In *EuroVis State-of-the-Art Reports* (2014).