

MultiLens: Fluent Interaction with Multi-Functional Multi-Touch Lenses for Information Visualization

Ulrike Kister, Patrick Reipschläger, Raimund Dachsel

Interactive Media Lab Dresden

Technische Universität Dresden, Germany

{firstname}.{lastname}@tu-dresden.de

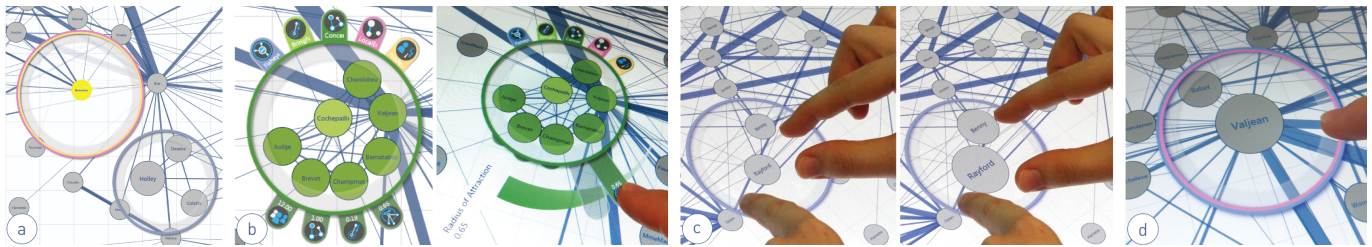


Figure 1. a) single and multi-function lenses; b) radial menu and slider for parameter adjustment; c) continuous gestures; and d) combination of lenses.

ABSTRACT

Interactive lenses have proven to be useful for many visualization applications where exploratory analysis is a primary task. Up to now, interaction with lenses is mostly limited to single-user, single-function lenses operated by mouse, keyboard and traditional parameter menus. To overcome these limitations, we propose *MultiLens*, touch-enabled magic lenses for fluently manipulating functions, parameters, and combinations of lenses on interactive surfaces. We contribute a novel multi-touch menu technique for magic lenses using a widget-based approach with a *drag-snap slider* for relative parameter adjustment. We also propose a *continuous gesture set* for rapidly changing lenses and their primary parameters in one seamless phrase. In addition, by supporting the combination of various lens functions, we create a generic multi-purpose lens tool. We illustrate our approach by investigating and implementing the concepts for the field of graph exploration. The prototype was evaluated in a user study with 22 participants comparing it to traditional parameter menus operated with both mouse and touch.

Author Keywords

Multi-touch interaction; information visualization; magic lenses; focus+context; graph exploration; graph lenses.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ISS'16, November 06 - 09, 2016, Niagara Falls, ON, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4248-3/16/11...\$15.00
DOI: <http://dx.doi.org/10.1145/2992154.2992168>

INTRODUCTION

Magic lenses [5] are established focus and context tools that have been applied to very diverse application domains. An important area where lenses are used is information visualization, where they can support exploration and understanding of very diverse information spaces and data types such as temporal or geo-spatial data. Magic lenses in visualization are defined as interactive tools temporarily altering a selected part of the data's representation [39]. The function of the lens can be an enlargement or distortion of content, like the most commonly used magnification lens, or other effects, e.g., to highlight or sample parts of the data. For this work, we use graph exploration as an exemplary visualization task as it is important in both research and practice: Graphs represent social networks for understanding social patterns, publication networks, or semantic ontologies. In biology, graphs visualizing metabolic pathways can improve comprehension of chemical processes in a cell. Examination and analysis of these networks can be valuable to very different users and domains. Lenses can help analyze these networks by highlighting relations, neighborhood, or patterns.

Many lens solutions have been proposed for data exploration, such as the RouteLens [1], the labeling lens [4], fisheye views for graphs [33], or the Bring Neighbors Lens [38]. However, many of them are limited to mouse-based positioning and global menus or dialog boxes for adjusting parameters. This decouples lens interaction from actual data exploration and creates discontinuities within the workflow. We expect large interactive surfaces to play a dominant role in both professional and private contexts. As soon as they are widely available, they can also be used for interactive exploration of information visualizations. Large interactive surfaces further lend themselves to collaborative data analysis supporting work of multiple people. With the advent of Natural User Interface techniques, novel ways of interacting with lenses have been proposed, among them multi-touch as a predominant input

modality. However, the full potential of touch-enabled magic lenses has not yet been unleashed, especially in the field of information visualization. So far, research has mostly focused on simply repositioning and resizing a lens [6, 20, 34]. However, this does not sufficiently utilize the power of interactive lenses. For their effective and efficient use for the exploration of visualizations, the lens function and its parameters have to be adjusted and manipulated as well. In this paper, we propose *MultiLens*, multi-purpose, multi-touch lenses that allow the application of different functions within the same lens. By synthesizing a collection of our own interaction concepts and existing principles, we contribute a functional multi-purpose magic lens system. We present multiple concepts for adjusting the lens function and its parameters and provide tools for their fluent adjustment using alternative manipulation techniques (see Figure 1). We contribute a widget-based approach with a *radial touch menu* for in-place interaction at the lens border that includes a novel *drag-snap slider* technique for efficient and fast manipulation of lens function parameters. We also propose a set of *continuous gestures* for both lens function activation and immediate adjustment of parameters in a seamless phrase. Additionally, our design allows the interactive combination of multiple lenses with different functions into one composite lens. While our general principles are applicable to other contexts, we implemented and evaluated our concepts for the specific context of graph exploration.

In the following sections, we will first provide an overview of related work focusing on magic lenses, their interaction and specific lenses for graphs. We will furthermore discuss our design of *MultiLens*, presenting both interaction techniques individually and then describe the combination of lenses. Finally, we report on a comparative study in which we evaluated our radial touch lens manipulation concept in contrast to global adjustment approaches with both mouse and touch.

RELATED WORK

There has been a variety of research on different aspects of magic lenses. While we focus on interaction, this paper is rooted in several of these earlier approaches. In the following, we will present related work in the context of magic lenses, their interaction and combination, as well as previous work in our example application context of graph exploration.

Magic Lenses and Defining the Lens' Selection

Magic lenses [5] are user interface tools of arbitrary shape, defining a focus region in which the representation of contained objects is altered, leaving the surrounding visualization as the context. As we cannot address all research on magic lenses, we refer to the survey by Tominski et al. [39] which presents a summary of data and tasks addressed by magic lenses in the context of visualization. It describes the three most important aspects of the lens: the *selection*, the *lens function*, and the *join* with the base visualization. Focus of most research on lenses are novel lens functions for specific use cases. Magnification is one obvious lens function in use, but reconfiguring, encoding, and filtering tasks can also be accomplished using lenses [39]. Contrary to the focus on developing new lens functions, in this work we advance the

interaction concepts for magic lenses. Instead of the visualization side, we focus on the often disregarded interaction side within information visualization applications [25].

The *selection*, i.e., the region of interest, is mostly defined by the position of the lens. For mouse interaction, an improvement on this aspect has been discussed by Appert et al. [2] who address target selection by taking the mouse cursor's input velocity into account. Similarly, Alvina et al. [1] improve interactions for the *lens' selection* by using the meta-data about routes on maps to counteract overshooting when moving the lens along a path. Next to position size, shape, and orientation also influence the *lens' selection* [39]. The labeling lens by Bertini et al. [4] improves exploration by automatically adjusting its size according to data density. Pindat et al.'s JellyLens [28] takes this further by dynamically adapting the shape of the lens to the content below.

When lenses are adjusted using the mouse, interaction is limited to subsequent changes of one parameter at a time, sometimes even on separate global widgets. Instead, multi-touch interaction allows for fast and convenient direct manipulation of elements.

Initially, lenses were used to improve touch or pen target acquisition using magnification. Ramos et al. [29] apply stylus triggered lenses for small target selection and conceptionally propose the adjustment of the zoom by direction-dependent crossing of the lens border. FingerGlass [20] presents fluent creation and re-positioning of lenses using touch for enhanced target selection which is made possible through bi-manual interactions. However, position and size were the only lens parameters adjusted. Similarly, Schmidt et al. [35] present a PushLens for graph exploration where only creation and re-positioning is possible. The lens-like telescope views of Khan et al. [21] include a fisheye magnification that can be moved and allows GUI-based adjustment of a zoom factor. Furthermore, Bortolaso et al. [6] apply magnification lenses to maps on interactive tabletops and a touch-enabled magnification lens for scatterplots has been introduced by Sadana and Stasko [31]. The PhysicLenses of Butscher et al. [7] further advance the use of lenses by applying multiple lenses to the same map data and discussing comparison possibilities. Smarties [11] uses magnification lenses for map data presented on a large display. These lenses' size and magnification factor can be remotely controlled by pinching on a mobile device.

All these examples apply multi-touch interaction to magic lenses for their specific use case and use repositioning and resizing to manipulate the *selection*. However, while the *lens' selection* is an important initial parameter that defines the data in focus, adjusting the lens function and its parameters is a second important aspect to make the lens effective to support the task and goals of the user. In previous work, parametrization was rarely used if at all, with the exception of a zoom factor. There are no solutions that accommodate flexible lens function parametrization and lenses with multiple functions. In this paper, we therefore primarily address the adjustment of the lens function and its parameters beside just basic repositioning.

Combination of Lens Functions

Existing lenses from related work often present one single lens function at a time. Yet, the combination of functions can improve efficiency [37] and create new lens functions [38]. The technical aspects of defining lenses to be able to compose them has previously been discussed by Fox [14]. Thiede et al. [37] focus on lenses as operators and base the possible combination on the Data State Reference Model. We use this as a foundation to create an interactive combination of lenses. For tangible interaction, combination of layered filters for image data has been presented in both embodied lenses [22] and TangibleRings [12]. However, these transparencies and rings are inflexible in size and cannot be easily parametrized.

Graph Interaction and Lenses for Graph Exploration

We selected graph exploration as our use case because it is a common task in information visualization which can be useful in a multitude of application cases, such as biology, social analysis, and database exploration. For graph exploration, Schmidt et al. [35] present a set of novel touch techniques for edge and connectivity exploration by plucking, pinning, strumming, and bundling edges, and Riche et al. [30] further discuss multi-touch gestures for control of edge bundling. To improve visibility in graph exploration, specific node magnification lenses have been developed as early as 1992 [32]. Carpendale et al. [9, 10] apply various parameters to magnification lenses, including different shapes, several fisheye functions and drop-offs for the effect. Tominski et al. [38] present a Bring Neighbors Lens that pulls in adjacent nodes of the ones in the *selection* for more information on the connectivity.

Multiple lenses have also been presented to counteract edge congestion, which is one major problem when exploring graphs. Wong et al. [41] propose an Edge Lens that uses distortion around a central point to push away all edges in proximity. The PushLens [35] takes this further by distorting edges around a circular lens area. The LocalEdge Lens [38] deals with local clutter by distinguishing relevant from irrelevant edges. Edges connected to nodes in the *selection* remain unchanged while all other edges crossing the lens are visually removed. Hurter et al. [19] apply different degrees of edge bundling and unbundling to edges in the *lens' selection*. Altogether there are other graph lenses for specific types of node-link diagrams and graph data [39], we only presented a selection of lens functions. While most of the lenses address graph exploration, few works like the EditLens [17] use them for manipulation.

Other than mouse and touch interaction, novel spatial interaction modalities were applied to lenses for the specific case of graph exploration. Spindler et al. [36] present semantic zooming on graph nodes through spatial movement of Tangible Views. Another approach was presented in BodyLenses [24] where the user's body is used to reposition and resize lenses on a display wall to explore a graph.

MAGIC TOUCH LENSES

As described in the previous section, lens manipulation in the past was mostly limited to positioning and resizing of lenses

using mouse (e.g., [1, 38]) or very basic movement of touch-enabled lenses (e.g., [6, 31, 35]). We believe that magic lenses as tools need to be able to be configured according to the users' needs to support them as best as possible in their current task and therefore need to support dynamically adjusting individual function parameters. We propose a touch interaction concept for magic lenses, called *MultiLens*, including manipulation and adjustment of the lens function, specifically focusing on fluent and seamless interaction. For that, we deliberately develop lenses as individual tools without global menus to keep the user's focus primarily on the selected data in a local context. This also supports the use on tabletops and wall-sized displays where global menus are not appropriate. In addition, multi-user scenarios are possible as lenses are created and manipulated independently of each other. Additionally, we designed the lens to resemble a ring to allow touch-through interaction to elements below the lens. This creates the significant advantage that elements within the region of interest can be further examined or manipulated.

As we address the combination of lenses, we distinguish the terms function and lens in this paper: A *lens* is the actual tool that can be manipulated. The *function* describes the effect or type of the lens, e.g., magnification or filtering. Each lens can have one or multiple functions with different parameters to consider. As it is the most commonly used form, our proposed lenses are of circular shape. The border is the visual representation of the lens and allows manipulation. As the border itself is rather thin to reduce occlusion of data, we widen the touchable area around it with a semi-transparent area. The color of the lens border encodes its current function. As multiple functions can be active in one lens, the border can consist of multiple colored rings (see Figure 1d).

Lens Functions and Parametrization

We use a set of graph-related lens functions as examples to illustrate our design. Our interaction concepts could also be applied to other data types and visualization techniques. However, this would require a different set of lens functions, e.g., ColorLens [13] for image data. Our set consists of the functions *Fisheye*, *Bring Neighbors*, *K-Neighborhood* and *Local Edge*. They were chosen for their diverse properties, addressing rendering, layout, or filtering of graph elements. Most research on lenses presented in prior work focus on the function itself, seldom offering the ability to dynamically change certain aspects of the function [39]. We analyzed these functions and identified a number of parameters that influence their effect. In the following, we describe our exemplary lens functions and their respective parameters.

The *Fisheye* function [10, 32] magnifies the content of the lens with continuous distortion toward the border. The power of the distortion can be altered by the repulsion parameter and the zoom level parameter changes the magnification. For the *Bring Neighbors* function [38], adjacent nodes are pulled in toward the lens. While the original concept only affected direct neighbors of the nodes in focus, we expanded it with a parameter controlling the degree of neighborhood at which nodes are affected. We call this the *K-Neighborhood* function. Nodes are arranged in concentric circles around a cen-

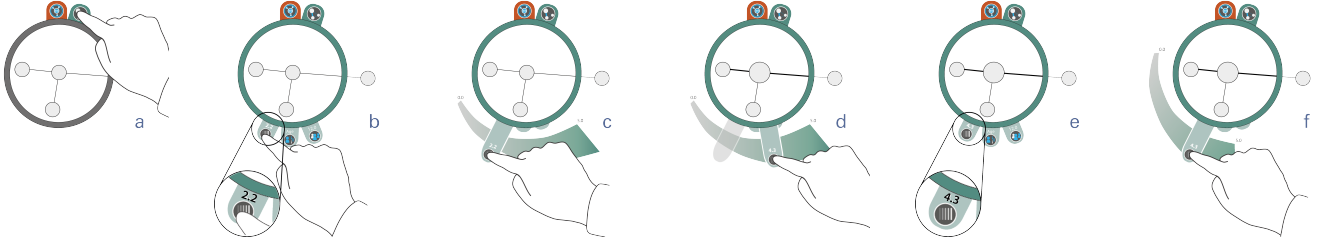


Figure 2. While the upper menu allows selection of desired lens functions (a), the lower handles allow parameter adjustments. The handle is *dragged* out (b-c) to adjust the value(c-d). On handle release, the handle *snaps* back to its original position (e). When pulled out again the position of the slider is automatically adjusted to the new value (f) again allowing relative manipulation.

ter node according to their degree (similar to Bring & Go by Moscovich et al. [27]), with an adjustable radius for the circles. The last parameter is the intensity with which the affected nodes are pulled towards the center node. The *Local Edge* function [38] declutters the view by removing crossing edges that are not connected to the selected nodes. We added a parameter which controls the transparency of the affected edges. Another parameter enables omitting all edges below a certain weight, with the transparency being only applied to the remaining edges.

Basic Manipulations of the Lens

We designed interactions for creating lenses as well as manipulating their position and size. Creation can be accomplished by holding five fingers in close proximity onto the interactive surface, similar to Smarties [11]. This gesture was chosen as the fingers form a circle which resembles the lens. The fingers provide a diameter as an initial lens size value. As the border embodies the lens as a whole, dragging on the lens border is used for translation of the lens, and scaling can be done using the well-known pinch gesture on the lens border. Furthermore, lenses can be removed by scaling them down to a certain minimal threshold which will create a fade-out effect and removes the lens if it is not resized or touched again.

Manipulation of Lens Functions and their Parameters

After being created, the generic lens has no active function yet. In this section, we present the *MultiLens* concepts to adjust a lens by selecting its function and by modifying function-dependent parameters. We first introduce a menu-based approach and then a gesture-only interaction technique.

Radial Touch Menu. To support novices or the infrequent use of *MultiLens*, we make the functionality explicit by using graphical widgets around the lens border, similar to GeoLens [40]. While GeoLens uses a fixed timescale around a circular element, we present a flexible tool for parameter adjustment. A radial context menu along the border of the lens exposes all possible functions and parameters to the user, enabling in-place interaction close to the data. Visibility of this menu can be toggled with a tap on the border. To support differentiation of lens functions and parameters, we separated the radial menu in two major groups. The handles at the top toggle the different lens functions, e.g., Fisheye or Local Edge, and are activated by a tap. Multiple active functions are supported, with the parameter handles for each function placed at the bottom of the lens. Parameter handles are only

visible when the respective function is activated. Each function has a unique color, which is used consistently for the lens border and all corresponding handles. For manipulating parameters, we propose two different interaction styles, one with sequential actions and the other with a seamless interaction phrase.

Discrete Tapping. Generally, a *discrete tap* on a parameter menu handle extends the handle and a slider appears around the lens. Simultaneously, other handles are miniaturized to make room (Figure 2c). Movement of the handle along the slider adjusts the parameter of the function. Parameters are either continuous with free handle movement or discrete where the handle snaps to defined steps. Another tap on the handle hides the slider again. We ensure that every parameter handle is always located at the same position on the lens border, using the user’s spatial memory to facilitate fast navigation.

Drag-Snap Technique. In our experience, function parameters are often not set to specific values but instead relative to the current value, e.g., bigger or smaller. For more efficient and fluent interaction, we propose a novel continuous technique for relative parameter adjustment on sliders called *drag-snap technique* as an alternative to discrete tapping. Parameter handles can be pulled out and smoothly dragged along the border of the lens to change a value in one continuous movement (Figure 2b-d). As soon as the finger is lifted, the handle snaps back to its original position on the lens border, selecting the chosen value. The slider automatically disappears while the selected value is still visible in the handle (Figure 2e). On dragging out the same handle again, the slider will have automatically adjusted its position (Figure 2f, cf. 2c). Therefore, relative adjustment is again possible. To summarize, if the user wants to change a parameter, she has to drag the handle outwards, move it to adjust the value, and release the touch.

In allowing both discrete, sequential tapping as well as smooth, continuous movement, we support inexperienced users as well as the transition to more fluent interaction for advanced users within the same widget-based approach. Additionally, all parameter handles are positioned at the bottom of the lens to allow observation of the current change to the visualization in the center of the lens while rotating the handle along the slider.

Continuous Gestures. To complement manipulating parameters using widgets, we propose a set of multi-touch gestures for the efficient and fast interaction with lenses. Together with the radial touch menu, *MultiLens* offers a joint interaction concept with different approaches to cater for different usage styles. Both allow triggering different functions within a single lens (see the accompanying video demonstrating the techniques). As tools are often immediately configured to suit the individual needs of a user, we developed continuous gestures allowing for a fluent transition from first activating the lens function to subsequently adjusting its primary parameter without lifting the finger. We thereby support an unbroken kinesthetic tension as previously discussed by Buxton [8] and recently presented by Gupta and McGuffin [18] for improved command selection and argument manipulation in radial menus.

To distinguish lens gestures from interaction with the underlying visualization, the user touches the lens' border with the non-dominant hand to set the frame of reference [3] for the interaction of the dominant hand. Besides decoupling gestures affecting the visualization from gestures affecting the lens, this also avoids complex gestures and provides easily remembered shortcuts for expert users. We present the following iconic gestures for the selected example lenses.

The *Fisheye* function is triggered using a pinch gesture (Figure 3a), which also allows seamlessly changing the degree of magnification. We employ this gesture as the resulting magnification is quite similar to zooming which is predominantly done using pinch-to-zoom gestures. For the classic *Bring Neighbors* function, affecting only direct neighbors, we deliberately designed the gesture mimicking the physical action: The user has to place her fingers onto the surface in form of a grabbing gesture, moving four or five fingers towards the lens center (Figure 3b). This also immediately manipulates the attraction radius. The *K-Neighborhood* function is triggered using a spiral drawing gesture (Figure 3c) which is similar to the concentric rings formed by the lens. Outward or inward spiral movement controls the degree of neighborhood, and the distance to the lens' center changes the radius of the circles. This gesture has also been examined in Cyclo-Zoom+ [26] where it was used for zooming purposes. Finally, triggering the *Local Edge* function is performed with a drag gesture resulting in a straight line from within the lens area towards the border of the lens, symbolizing dragging an edge away from the lens area (Figure 3d). Transparency of the edges is directly controlled by the distance of the touch to the lens' center.

To summarize, the proposed gestures provide a seamless transition from activation to adjustment of the lens function, seamlessly phrasing together individual interaction steps [8].

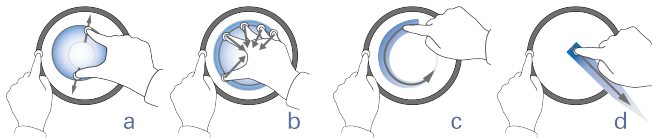


Figure 3. A recognized gesture switches the lens function while the continuation of the gesture allows for adjustment of a primary parameter.

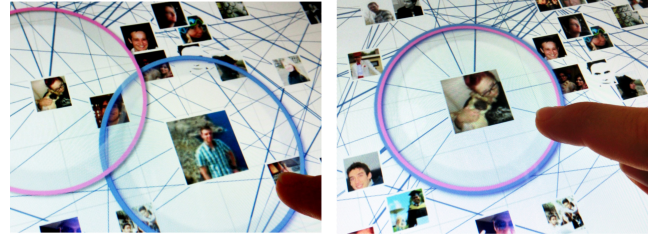


Figure 4. When lenses of about the same size overlap, they can be combined to form one lens. Here both *Fisheye* (blue) and *Local Edge* (pink) function are active.

Experts can trigger functions efficiently and fluently adjust the main parameters within their workflow. Similar to Frisch et al. [16], with *MultiLens* we advocate the creation of interaction techniques for both novices and experts, emphasizing the need for effective multi-touch techniques for fluent use. To support all users, a switch between gesture and menu-based interaction is possible at any time without explicit mode switches.

Combination of Lenses

Similar to the presented parameter adjustment of lenses, *MultiLens* supports alternative ways of combining multiple lens functions. Moving two lenses on top of each other applies their individual lens functions onto the shared selection. This is a very clear interaction for novice users. If both lenses are similar in size (below a certain threshold) and would cover the same area, a colored ring inside the lens indicates that the lenses may be combined. Releasing the touch when feedback is visible confirms the combination. The feedback vanishes when the lens is held for a second, keeping the lenses from snapping together. Lenses of different size can be moved on top of each other without automatic combination. A combination of *Fisheye* and *Local Edge* function can be seen in Figure 4. This combination presents a new tool in itself that supports focusing on central nodes. It can be moved as one lens. Separation of lenses can be accomplished by holding a finger on the border of the lens. This will split the lens border into individually colored segments representing the currently active functions which can then be dragged out from the lens to be removed.

While this interaction for combining lenses requires two lenses already in place, simply adding a function to an existing lens can also be useful. Therefore, we distinguish two modes when using the expert gestures: replace or add. These modes are differentiated by the number of fingers placed on the lens border with the non-dominant hand, again setting the frame of reference for the interaction. We use two fingers to activate the alternative action, similar to [15]. Performing the gesture with two fingers on the lens border will add the new function to the lens instead of replacing the existing one(s).

EVALUATION: COMPARISON STUDY

We conducted a user study to assess our *MultiLens* concept aiming at gaining insights for further iterations as well as evaluating its usability and suitability for practical use. Since the flexible radial menu including the *drag-snap slider technique* is a central aspect of our approach, we investigated its

comprehensibility and efficiency. In this work, we focus on the basic approach suitable for all users and leave the study of the advanced gestures for experts for future work. We decided to compare the radial menu concept to state-of-the-art implementations of magic lenses. Since there are no comparable solutions on large displays, we use a single-user single-display setup. Most state-of-the-art lens implementations use mouse interaction and global menus or dialogs for the parameter adjustments [39]. Therefore, we use a traditional side menu for comparison. However, to distinguish between changes resulting from the interaction modality, we decided to evaluate both touch and mouse interaction for the traditional menu. This results in three conditions: mouse + traditional menu (M+TM), touch + traditional menu (T+TM), and our own approach touch + radial menu (T+RM).

We hypothesized that the radial touch menu (T+RM) as part of our *MultiLens* is more efficient in terms of required time per action than the current state-of-the-art menu versions (both M+TM and T+TM) because of shorter distances between lens and menu. Furthermore, we believed that users will prefer the directness of the novel *MultiLens* implementation over the state-of-the-art as interaction is less decoupled from the focus (i.e., the selected data).

Participants. We recruited 22 unpaid participants (all right handed, 5 female) to take part in this comparative evaluation study. The participants were students of the math, computer science, and engineering departments of our university. Their age ranged from 21 to 36 ($M = 24.7$, $SD = 3.3$). All but two reported daily use of mobile touch-enabled devices, of which eight had occasional to frequent use of larger touch-surfaces (interactive white-boards or wall-sized displays) and eight had never worked with touch on devices larger than tablets. 13 had no or very little experience with graphs, while nine stated they work with graphs occasionally, e.g., for UML diagrams or research projects. Magic lenses were completely unknown or never used by 14, three had little experience and only five had worked with magic lenses occasionally up to frequently for research or student projects.

Apparatus. We implemented our *MultiLens* concept in a C#/WPF application for graph exploration (extended from our conference demo [23]). We also created a state-of-the-art implementation where lenses can be dragged and selected on their border. To manipulate their function and parameter, we also designed a side menu with WPF menu elements for use with either touch or mouse interaction. The touch conditions (both T+TM and T+RM) ran on a 27" Perceptive Pixel¹ display with a resolution of 2560x1440 which was slightly tilted for comfort. For M+TM, we used a standard vertical PC display setup with the same resolution and size to represent a state-of-the-art setup for comparison (see Figure 5). Both displays were connected to the same PC.

Methodology and Procedure. The time used for the experiment per participant was approx. 50 min. Prior to data collection, participants completed a questionnaire on their demographic details and experience. All participants were in-



Figure 5. The study was conducted using a slightly tilted display for both touch conditions and a vertical PC display for the mouse condition. The traditional menu (right) was used both with mouse and touch for comparison with our radial touch menu.

troduced to the effect of each lens function that was used in the study by being presented with images on paper. Afterwards, they were instructed to make themselves comfortable in front of either the PC display (for mouse condition) or the Perceptive Pixel display (for touch conditions). The experiment was a within-subject design and the order of conditions was counterbalanced for 21 participants. An additional 22nd participant was recruited to replace an outlier (see observations below).

For each condition the participant started with a training phase where a researcher explained the basic functionality of each menu, starting with the basic repositioning of the lens to individual lens function parameter adjustments. The participants were given time to make themselves familiar with the interactions until they felt comfortable using the condition ($M = 3.1$ min). They could then trigger the beginning of the experiment. In each task, participants were presented with the task description and they had time to carefully read the instructions before pressing the start button and moving to the graph. We created three social graphs with random names as the study's data sets. To balance complexity, the number of elements (77 nodes, 254 edges) and node degrees were the same. Using a force-directed layout, each graph was arranged differently. They were counterbalanced so that each graph was used equally often among conditions.

We designed the experiment to include two groups of tasks: six *process tasks* (A) and three *exploration tasks* (B). In the first group (A), for each task participants were given a list of specific instructions stating up to four individual manipulation steps. Examples include (1) "Trigger lens function Local Edge", (2) "Set function parameter Edge Opacity to 0.2", etc. We designed the process tasks to conform with normal lens exploration processes. Hence, the tasks often started or ended with a move of the lens onto a specific, highlighted node. For *process tasks*, we measured task completion times and logged times of individual interaction steps. *Exploration tasks* (B) were less specific in which interaction steps to take. They included questions such as "Find and select the vegetarian among the friends of Frida" or "Select all neighbors of Lorenzo". We designed each graph for these tasks to include the same problems (e.g., edge clutter, so that the node in question had unconnected edges running through and using the *Local Edge* function was necessary). We also logged completion times for these tasks, but their exploratory char-

¹<http://www.perceptivepixel.com>

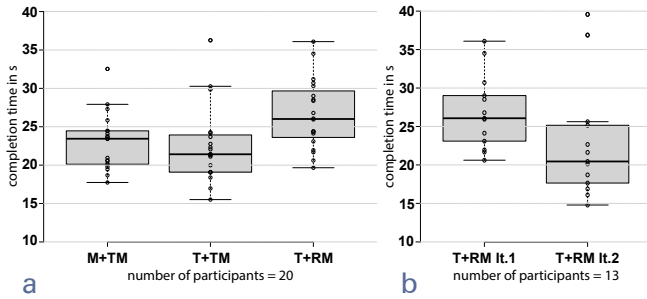


Figure 6. Box plots using quartiles with Tukey whiskers (created with <http://boxplot.tyerslab.com>). a) T+RM is significantly, albeit in regard to absolute time only slightly slower than M+TM and T+TM. However, b) the second iteration of T+RM is significantly faster than the first one.

acter makes it difficult to tell how much of the time was affected by the menu technique. These tasks primarily served for observing the exploration process. Furthermore, we video recorded the interactions of each user for later analysis and took notes of our observations during the task completion.

After every condition, the participants filled out a separate questionnaire of NASA TLX questions with a 7-point scale to evaluate the physical and mental demand and stated the perceived ease while manipulating the lens. After completing all three conditions, the users were asked to complete a final comparing questionnaire, checking which of the conditions was perceived as most and least helpful, fastest, most comfortable in use, and applicable in everyday work.

Observations and Results

The analysis of the completion times for the process tasks showed outliers for two participants who we also noticed had problems reading the descriptions or partially disregarded the instructions. We removed those two participants for all conditions. As a Shapiro-Wilk test indicated that the assumption of normal distribution was violated, we used Friedman's ANOVA showing a significant difference between T+RM and the other conditions ($\chi^2(2) = 17.1, p \leq .001$). There was no significant difference between M+TM and T+TM. Even though we hypothesized a better performance, T+RM ($M = 26.59$ s) was slightly slower than M+TM ($M = 22.95$ s) and T+TM ($M = 22.45$ s) (see Figure 6a). We hypothesize that a potential reason for this result was the novelty of the menu and the little experience most participants had with larger displays and magic lenses. Therefore, we asked 13 of the participants to perform an additional repetition of the six process tasks using T+RM to evaluate potential learning effects. We statistically analyzed the difference between their first and their second iteration with T+RM (normally distributed, ANOVA with Bonferroni correction). These participants were significantly faster ($F_{1,12} = 13.39, p \leq .05$) in the second run ($M = 22.79$ s), with an average of 15.2% increase in speed (see Figure 6b). This shows that experience is an essential factor in this scenario. While with further training all conditions will likely improve, the interaction principles of traditional menus are well-established, so that the strong improvements with the novel condition suggest that the differences between conditions might even disappear.

The evaluation of the NASA TLX questionnaires found no significant differences (ANOVA) between the conditions regarding mental or physical demand, effort, or frustration. While no assertion can be made, we see no evidence for a difference in perceived demand between techniques. In the final questionnaire, the participants were asked to compare the three conditions. 62% of participants selected T+RM as the condition they felt supported them best. Comfort of use was also considered best with T+RM by 57% of participants. When asked to guess the fastest of the conditions, 43% of the participants selected T+TM, 33% T+RM and only 24% checked M+TM. Although we tried to hide which of the conditions were our own design, we cannot rule out a possible bias due to the novelty effect which might have influenced the participant's selection.

Beside the questionnaire, we used the video data and our own observations during both the *process* but also especially the *exploration tasks* to further assess the participant's interactions, concentrating on our T+RM technique to gain insight on how participants worked with the lenses. One aspect of T+RM, which we frequently observed, was that participants had problems identifying parameters based on their icons. They needed to expand the handle to be able to read the label, often resulting in trying out several handles before discovering the right one. This may have been partially responsible for the slower task completion times.

Participants were offered free choice between using tap or drag-snap for manipulating parameters. We observed that of the 22 participants, 14 mainly used the *drag-snap technique*, only three used *discrete tapping* primarily and five participants frequently switched techniques. We noticed that using tap often coincided with situations where participants were unsure on how to proceed or which parameter to select. This assumption is further supported by the questionnaires, where the three users of *discrete tapping* belonged to the group which had little to no experience with large touch displays, graphs, and magic lenses. In all three conditions, some tasks required using two lenses while other tasks provided two lenses that could be used freely. We noticed that when given the choice, 16 of 22 participants used only a single lens, combining various lens functions instead of assigning a unique function to each lens.

DISCUSSION

In this work, we focused on a multi-purpose lens design which is capable of unifying multiple functions in one lens tool. Even though *MultiLens* was slower in our study, time differences in terms of overall time were generally small, and our evaluation shows that more experience and training results in improved efficiency. Furthermore, contrary to traditional menus, *MultiLens* is applicable to large displays and multi-user scenarios because of its support for in-place manipulation and local adjustment of the tool. With all interaction taking place directly around the explored data, minimizing the distraction resulting from gaze switches, the user can better immerse herself in her current task.

Our study uncovered a flaw in our design as our participants had to expand handles of individual function parameters to

reveal their labels when the handle icon itself was not sufficient. This needs to be addressed in a future iteration and is especially important for domain novices, which are still in the learning process and therefore unsure about individual parameters. With the increased use of multi-touch for more complex functionality and without the help of tooltips, there is a heightened need for a diverse set of well-designed icons. This results in a challenge for HCI designers to improve the distinguishability of parameters, making them easily recognizable even for novice users. Short textual labels can complement the icons and improve recognition.

Furthermore, our study indicated that users prefer to extend the functionality of a single lens instead of using multiple independent lenses. This observation strengthens our approach of supporting flexible combination of multiple lens functions into a single tool. Nonetheless, it raises the question to further research if single adjustable tools with rich, complex functions are to be preferred to multiple single purpose tools. However, one limitation of the radial menu is that the space for menu handles is dependent on the size of the lens, which limits the possible number of lens functions and parameters. This needs to be investigated further.

An important contribution of our work is the flexible parametrization and combination of lens functions. Related work on touch-enabled lenses [6, 11, 20, 31] focused mostly on magnification, only sometimes allowing the adjustment of a zoom factor. Their interaction principles were relatively limited and did not support flexible lens functions nor their parametrization. However, we believe lenses can be a rich tool applicable to many use cases with diverse lens functions. Their combination further encourages this toolbox-like character. Nevertheless, to support these functionalities additional interaction concepts are necessary. We hence took existing related work and extended it by applying lenses to a more advanced use case that goes beyond simple magnification of content. As a result, our approach supports both the combination of lenses as well as the individual parametrization of lens functions. Our interaction concepts are strongly integrated into the overall approach of the lens, allowing manipulation in-place, and – as we observed in the study – were found pleasant to use.

In our prototype, we used *MultiLens* for graph exploration. Nevertheless, our interaction design principles are also applicable to other visualizations, e.g., examination of medical images by applying additional image filters. We think that professional tasks like these will soon be accomplished using natural user interfaces on mobile devices and large interactive surfaces. However, appropriate gestures for these other domains need to be designed. An advantage of our system are the diverse, redundant interaction possibilities that can be used to trigger functionality. Participants in our study switched between techniques (discrete tapping, continuous drag-snap) depending on context, confidence, and experience. In joining these menu and slider techniques with our continuous gestures, we designed a flexible system, offering easy adjustment of functions and parameters while supporting personally preferred interaction styles. We think that providing

tools with both smooth widget-based and gestural interaction capabilities yields improved interaction and has potential for efficiently supporting the work of a larger range of users. These principles can be generalized and extended to other designs where tool creation and parameter adjustments are in focus. Especially for information visualization, interactive surfaces can improve interaction by providing tools that allow casual exploration and direct feel of the data while also providing precise input capabilities for efficiency. By fluently combining and adjusting lens functions using multi-touch, we advocate a novel way of more direct interaction with information visualizations.

CONCLUSION

We contributed *MultiLens*, novel multi-touch-enabled visualization lenses that provide multiple lens functions within one generic lens tool. As previous work focused on single-function lenses, we apply existing principles and improve on the interaction with lenses considering a) basic interactions, b) activation of different functions, c) the adjustment of function-dependent parameters, and d) the combination of lens functions. In a comparative evaluation, we found slower task completion compared to traditional menus with mouse and touch. However, further training resulted in significant performance improvements. In contrast to traditional approaches, our lenses support seamless in-place manipulations and adjustments which contribute to the user's flow of interaction. We proposed multiple ways of function and parameter adjustments presenting both a widget-based approach using a novel *drag-snap technique* and a set of *continuous gestures*. Both enable fluent interaction in uninterrupted phrases. Furthermore, the combination of lenses yields entirely new lens functions. With this, we contributed another step in using the potential of interactive surfaces for information visualization by presenting a flexible tool with different interaction alternatives.

For future work, we plan to apply our *MultiLens* concept to other visualization domains and their lens functions. This would offer the possibility for in-depth evaluation of our extended approach together with domain experts. Additionally, even though *MultiLens* already supports basic multi-user scenarios, we want to further investigate their special requirements and the necessary additions to our concepts.

ACKNOWLEDGMENTS

The authors would like to thank Mathias Frisch and the students Martin Zenker, Elisabeth Wienß, and Sara Groß for their involvement in this project. This project was in part funded by the DFG (DA 1319/3-1) and the Excellence Initiative of the German Federal and State Governments (Institutional Strategy, measure “support the best”).

REFERENCES

1. Alvina, J., Appert, C., Chapuis, O., and Pietriga, E. Routelens: Easy route following for map applications. In *Proc., AVI '14*, ACM (2014), 125–128.
2. Appert, C., Chapuis, O., and Pietriga, E. High-precision magnification lenses. In *Proc., CHI '10*, ACM (2010), 273–282.

3. Balakrishnan, R., and Hinckley, K. The role of kinesthetic reference frames in two-handed input performance. In *Proc., UIST '99*, ACM (1999), 171–178.
4. Bertini, E., Rigamonti, M., and Lalanne, D. Extended excentric labeling. In *Proc., EuroVis'09*, The Eurographs Association (2009), 927–934.
5. 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.
6. Bortolaso, C., Oskamp, M., Phillips, G., Gutwin, C., and Graham, T. N. The effect of view techniques on collaboration and awareness in tabletop map-based tasks. In *Proc., ITS '14*, ACM (2014), 79–88.
7. Butscher, S., Hornbæk, K., and Reiterer, H. Spacefold and physiclenses: Simultaneous multifocus navigation on touch surfaces. In *Proc., AVI '14*, ACM (2014), 209–216.
8. Buxton, W. A. S. Human-computer interaction. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, ch. Chunking and Phrasing and the Design of Human-computer Dialogues, 494–499.
9. Carpendale, M. S. T., and Montagnese, C. A framework for unifying presentation space. In *Proc., UIST '01*, ACM (2001), 61–70.
10. Carpendale, S., Ligh, J., and Pattison, E. Achieving higher magnification in context. In *Proc., UIST '04*, ACM (2004), 71–80.
11. Chapuis, O., Bezerianos, A., and Frantzeskakis, S. Smarties: An input system for wall display development. In *Proc., CHI '14*, ACM (New York, NY, USA, 2014), 2763–2772.
12. Ebert, A., Weber, C., Cernea, D., and Petsch, S. Tangiblerings: Nestable circular tangibles. In *Ext. Abstracts CHI*, CHI EA '13, ACM (2013), 1617–1622.
13. Elmqvist, N., Dragicevic, P., and Fekete, J.-D. Color lens: Adaptive color scale optimization for visual exploration. *IEEE Trans. Vis. Comput. Graphics* 17, 6 (June 2011), 795–807.
14. Fox, D. Composing magic lenses. In *Proc., CHI '98*, ACM (1998), 519–525.
15. Frisch, M., Heydekorn, J., and Dachzelt, R. Diagram editing on interactive displays using multi-touch and pen gestures. In *Proc., Diagrams '10*, Springer (Aug. 2010), 182–196.
16. Frisch, M., Langner, R., and Dachzelt, R. Neat: a set of flexible tools and gestures for layout tasks on interactive displays. In *Proc., ITS '11*, ACM (2011), 1–10.
17. Gladisch, S., Schumann, H., Ernst, M., Füllen, G., and Tominski, C. Semi-automatic editing of graphs with customized layouts. *Computer Graphics Forum* 33, 3 (June 2014), 381–390.
18. Gupta, S., and McGuffin, M. J. Multitouch radial menu integrating command selection and control of arguments with up to 4 degrees of freedom. In *Proc., AVI '16*, ACM (2016), 256–263.
19. Hurter, C., Telea, A., and Ersoy, O. Moleview: An attribute and structure-based semantic lens for large element-based plots. *IEEE Trans. Vis. Comput. Graphics* 17, 12 (2011), 2600–2609.
20. Käser, D. P., Agrawala, M., and Pauly, M. Fingerglass: efficient multiscale interaction on multitouch screens. In *Proc., CHI '11*, ACM (2011), 1601–1610.
21. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., and Kurtenbach, G. A remote control interface for large displays. In *Proc., UIST '04*, ACM (2004), 127–136.
22. Kim, K., and Elmqvist, N. Embodied lenses for collaborative visual queries on tabletop displays. *Information Visualization* 11, 4 (Oct. 2012), 319–338.
23. Kister, U., Reipschläger, P., and Dachzelt, R. Multi-touch manipulation of magic lenses for information visualization. In *Proc., ITS '14*, ACM (2014), 431–434.
24. Kister, U., Reipschläger, P., Matulic, F., and Dachzelt, R. Bodylenses: Embodied magic lenses and personal territories for wall displays. In *Proc., ITS '15*, ACM (2015), 117–126.
25. Lee, B., Isenberg, P., Riche, N. H., and Carpendale, S. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Trans. Vis. Comput. Graphics* 18, 12 (2012), 2689–2698.
26. 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.
27. Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., and Fekete, J.-D. Topology-aware navigation in large networks. In *Proc., CHI '09*, ACM (2009), 2319–2328.
28. Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. Jellylens: content-aware adaptive lenses. In *Proc., UIST '12*, ACM (2012), 261–270.
29. Ramos, G., Cockburn, A., Balakrishnan, R., and Beaudouin-Lafon, M. Pointing lenses: Facilitating stylus input through visual-and motor-space magnification. In *Proc., CHI '07*, ACM (2007), 757–766.
30. Riche, N. H., Dwyer, T., Lee, B., and Carpendale, S. Exploring the design space of interactive link curvature in network diagrams. In *Proc., AVI '12*, ACM (2012), 506–513.
31. Sadana, R., and Stasko, J. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proc., AVI '14*, ACM (2014), 265–272.

32. Sarkar, M., and Brown, M. H. Graphical fisheye views of graphs. In *Proc.*, CHI '92, ACM (1992), 83–91.
33. Sarkar, M., and Brown, M. H. Graphical fisheye views. *Commun. ACM* 37, 12 (Dec. 1994), 73–83.
34. Schmidt, D., Chong, M. K., and Gellersen, H. Idlenses: Dynamic personal areas on shared surfaces. In *Proc.*, ITS '10, ACM (2010), 131–134.
35. Schmidt, S., Nacenta, M. A., Dachzelt, R., and Carpendale, S. A set of multi-touch graph interaction techniques. In *Proc.*, ITS '10, ACM (2010), 113–116.
36. Spindler, M., Tominski, C., Schumann, H., and Dachzelt, R. Tangible views for information visualization. In *Proc.*, ITS '10, ACM (2010), 157–166.
37. Thiede, C., Fuchs, G., and Schumann, H. Smart lenses. In *Smart Graphics*, vol. 5166 of *SG '08*. Springer-Verlag, 2008, 178–189.
38. 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 (2006), 17–24.
39. Tominski, C., Gladisch, S., Kister, U., Dachzelt, R., and Schumann, H. Interactive lenses for visualization: An extended survey. *Computer Graphics Forum* (2016).
40. von Zadow, U., Daiber, F., and Schöning, J. GeoLens: Multi-User interaction with rich geographic information. *Workshop DEXIS* (2012), 16–19.
41. Wong, N., Carpendale, S., and Greenberg, S. Edgelens: an interactive method for managing edge congestion in graphs. In *Proc.*, InfoVis'03, IEEE Computer Society (2003), 51–58.