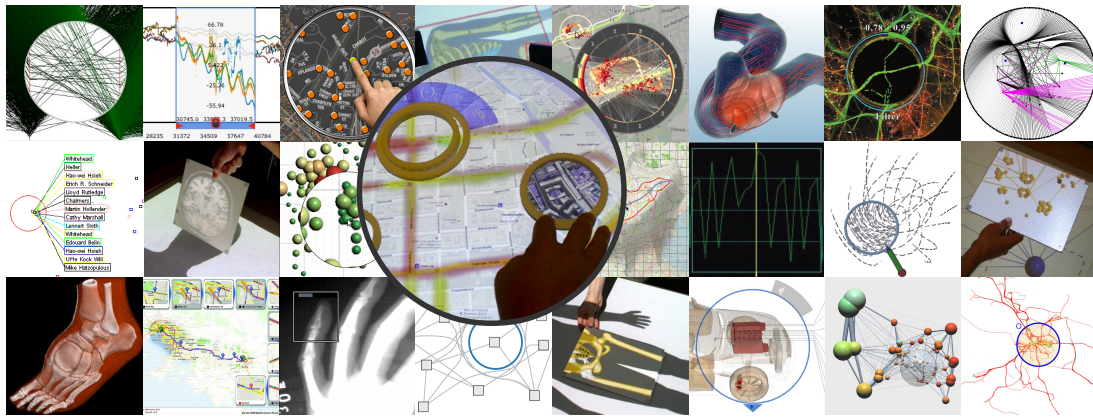


# Interactive Lenses for Visualization: An Extended Survey

C. Tominski<sup>1</sup>, S. Gladisch<sup>1</sup>, U. Kister<sup>2</sup>, R. Dachsel<sup>2</sup> & H. Schumann<sup>1</sup>

<sup>1</sup>Institute of Computer Science, University of Rostock

<sup>2</sup>Interactive Media Lab, Technische Universität Dresden



**Figure 1:** A mosaic of interactive lenses being applied in visualization contexts.

---

## Abstract

*The elegance of using virtual interactive lenses to provide alternative visual representations for selected regions of interest is highly valued, especially in the realm of visualization. Today, more than 50 lens techniques are known in the closer context of visualization, far more in related fields. In this article, we extend our previous survey on interactive lenses for visualization. We propose a definition and a conceptual model of lenses as extensions of the classic visualization pipeline. An extensive review of the literature covers lens techniques for different types of data and different user tasks and also includes the technologies employed to display lenses and to interact with them. We introduce a taxonomy of lenses for visualization and illustrate its utility by dissecting in detail a multi-touch lens for exploring large graph layouts. As a conclusion of our review, we identify challenges and unsolved problems to be addressed in future research.*

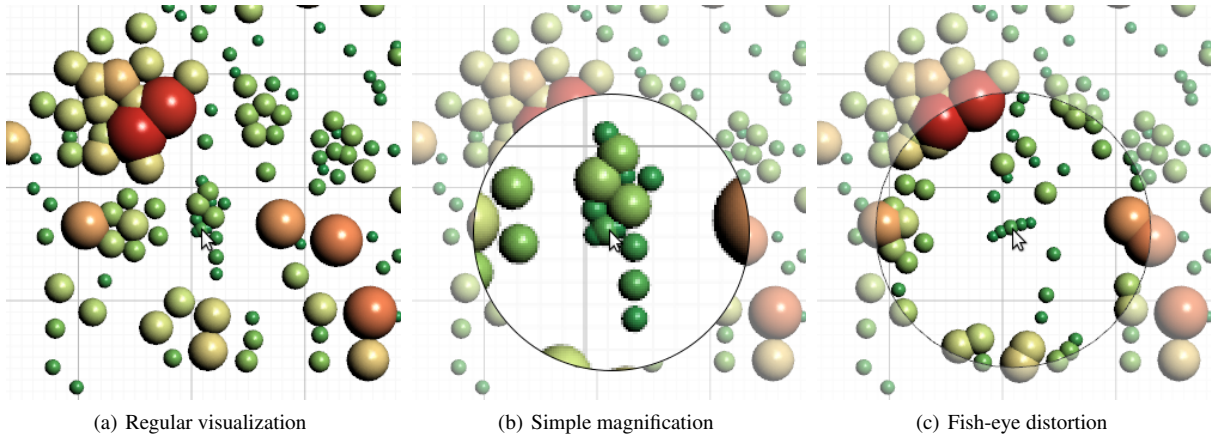
Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction Styles

---

## 1. Introduction

Visualization has become an established means to help people in various fields to gain insight into their data [WGK10]. Yet, as data size is constantly increasing, visualization approaches have to deal with the problem of cluttered and overcrowded visual representations. Moreover, the complexity of the data makes it difficult to encode all relevant information contained in a dataset into a single visualization image.

Addressing these challenges, visualization researchers utilize concepts such as overview+detail and focus+context [LA94, Hau06, CKB08] as well as multiple coordinated views [Rob07], which facilitate the exploration of large datasets and provide multiple perspectives on complex data. The idea is to enable the user to interactively change the perspective on the data very much in the sense of what Bertin [Ber81] said:



**Figure 2:** The regular visualization (a) with a cluster of cluttered dots in the center. A simple magnification (b) shows a larger picture, but cannot untangle the cluster. A fish-eye distortion (c) helps in clearing the view on individual dots in the cluster.

“A graphic is not ‘drawn’ once and for all; it is ‘constructed’ and reconstructed until it reveals all the relationships constituted by the interplay of the data. The best graphic operations are those carried out by the decision-maker himself.”

Interactive lenses are an important class of methods to support interactive multi-faceted data exploration. The idea behind interactive lenses is to provide on demand an alternative visual representation of the data underlying a local area of the screen. This general concept is as simple as it is powerful and versatile. In early works on *magic lenses*, Bier et al. envisioned several possible instances of useful lenses, including lenses for magnifying visual items, adjusting graphical properties, querying precise data values, or dynamic filtering [BSP\*93, BSF\*94, SFB94, FS95, BSP97].

The visualization community with its need for flexible interactive tools to support the exploration of complex data has embraced the idea of interactive lenses. In the last two decades more than 50 lenses have been proposed in the closer context of visualization research, still more in related fields that deal with visual information as well (e.g. human-computer interaction or augmented reality). In this work, we survey the rich body of literature on lenses in visualization, of which Figure 1 illustrates only a subset.

**Introductory Visualization Example** Let us start with a simple example of a classic magnification lens. Real-world magnification lenses have an ancient history as tools allowing us to look at details that cannot be seen with the human eye alone. The need to look at details has also motivated early work on overview+detail, focus+context, and magic lenses [Fur86, MRC91, BSP\*93, LA94].

An interactive magnification lens is typically of circular shape and positioned on the screen where a more detailed

view is needed. The lens will magnify the content underneath the lens according to a mathematical specification, modifying either the pixels of the display, the geometric model, or even the underlying data.

But what on first sight appears to be rather straightforward turns out to be the result of carefully made design decisions. Consider the visualization depicted in Figure 2(a), which shows dots whose size and color visually encode numerical data values. The dots in the center near the mouse pointer occlude one another making it impossible to identify them individually. A simple magnification of the image’s pixels as shown in Figure 2(b) allows us to acquire a bigger picture of the situation, but it does not help to untangle the occlusion. Applying magnification to the geometric model to be rendered will not help either, because it would lead to the same visual output.

Instead a lens is needed that magnifies only the distances between dots, while the dots’ sizes are left unchanged. As shown in Figure 2(c), realizing such a lens via a fish-eye transformation helps to clear the view on the dots in a local region. Of course such a temporary distortion is allowed only if the dots’ positions bear no information that is critical to the task at hand.

Figures 2(b) and 2(c) also illustrate the visual feedback crafted to communicate the fact that a lens is in effect. A circle makes the user aware of the geometric properties of the lens (i.e., shape, position, and size). Further, a virtual veil has been cast on those parts of the visualization that are not affected by the lens, effectively directing the user’s attention to the content presented within the lens.

Our introductory example indicates that interactive lenses can be quite useful as lightweight tools to resolve problems encountered during data exploration. The example also suggests that there is more to lenses than simple magnification

of image pixels. The next sections will elaborate on the manifold designs, purposes, and implementations of interactive lenses in visualization.

**Outline** In Section 2, we start with describing how lenses are defined and how they are integrated into the visualization process. Keeping our considerations on an abstract level, we establish a conceptual model that combines the classic visualization pipeline and interactive lenses. Key points are to define where and to what extent a lens takes effect, how the effect is generated, and how the visual feedback is fetched back to the visualization. Descriptions of basic properties of lenses and discussions of further aspects complete the conceptual part.

Concrete examples of lens techniques will be given in Section 3. We present a structured view on interactive lenses in visualization along two key aspects of visualization: *the data* as the subject of interest and *the task* as the objective of the user. We use a set of data types inspired by Shneiderman’s taxonomy [Shn96] to categorize lenses with respect to the data aspect. Yi et al.’s [YKSJ07] categories of user intents for interaction serve to structure lenses according to the task they support.

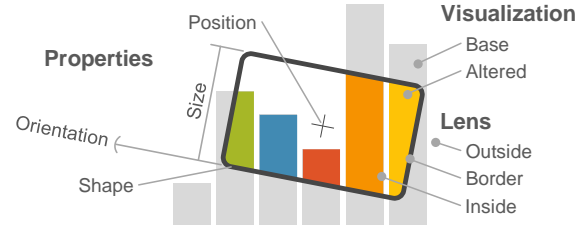
In Section 4, we consider the display devices and the interaction modalities that are employed to show lenses (and visualization) and to interactively control them. While many lenses work with a classic desktop setup with regular display and mouse plus keyboard, there are modern technologies (e.g., large high-resolution displays, multi-touch interaction, head-tracking) with new capabilities that can be utilized to make interactive lenses even more useful in visualization.

The content of Sections 1 to 4 are revised and improved versions of our earlier survey on interactive lenses [TGK\*14]. We expanded the discussion of the conceptual model, included the most recent publications on lenses in visualization, added more illustrating figures, and improved the quality of the existing figures.

Completely new in this extended survey is the taxonomy for lenses in visualization in Section 5. The taxonomy is directly extracted from the collected knowledge of our literature survey. With the help of the taxonomy, existing and new lens techniques can be organized according to practical and conceptual aspects.

The utility of the taxonomy is illustrated with a comprehensive example in the new Section 6. We dissect in detail a graph lens operating on a touch-enabled desktop display and summarize all aspects as covered by our taxonomy.

Moving on from existing lens techniques, we will take a look into the future in Section 7. We describe prospects of interactive lenses in visualization and identify corresponding challenges to be addressed in the research to come. A brief summary in Section 8 concludes our extended survey.



**Figure 3:** Schematic depiction of an interactive lens.

## 2. Interactive Lenses for Visualization

Lens techniques are lightweight on-demand tools. As illustrated in Figure 3, a lens is a parameterizable selection according to which a base visualization is altered. Typically, a lens is added to a visualization interactively to solve a specific localized problem. A key characteristic is that the lens effect (i.e., the modification of the visualization) is transient, that is, the visualization returns to its original state once the lens is dismissed.

Originally, the visible effect of a lens was locally confined to the lens interior [BSP\*93]. In the context of visualization, there are less strict interpretations of what a lens is. A lens technique might affect the visualization beyond the confines of the lens or even show the lens and its effect separately. Our survey also includes such techniques as long as they have a lens-like feel or call themselves lenses.

According to these lines of thinking, we define lenses for visualization as follows:

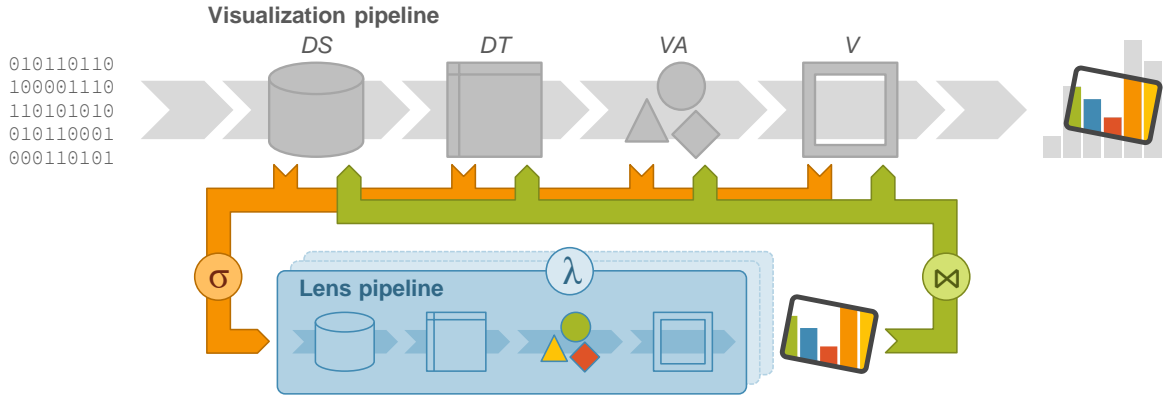
**Definition** *An interactive lens is a lightweight tool to solve a localized visualization problem by temporarily altering a selected part of the visual representation of the data.*

Closely related to lens techniques are illustrative visualization techniques such as cutaway views [FS92], exploded views [BG06], or peel-away views [CSC06]. Moreover, there are some overview+detail and focus+context techniques that influence the display globally [LA94]. While they are not in the focus of our survey, we still include selected references to such techniques in the appendix.

Next in Section 2.1, we will introduce a conceptual model for lenses in the context of visualization. In Section 2.2, we take a look at general properties of lenses.

### 2.1. Conceptual Model

To define a conceptual model of interactive lenses for visualization, we build upon the well-known visualization pipeline [CMS99]. The visualization pipeline describes how data is transformed from a data source (*DS*) via data tables (*DT*) and visual abstractions (*VA*) to a visualization view (*V*). We consider lenses in relation to this pipeline and its various stages of transformations.



**Figure 4:** Conceptual model of lenses. A lens pipeline implements a lens function  $\lambda$  to generate a lens affect. The lens pipeline is attached to a standard visualization pipeline via a selection  $\sigma$  and join  $\bowtie$ .

Figure 4 shows that a visualization lens can be modeled as an additional lens pipeline that is attached to a standard visualization pipeline. The lens pipeline realizes a *lens function*  $\lambda$  that generates a lens effect. There are two points of information exchange between the standard pipeline and the lens pipeline. The first is a *selection*  $\sigma$  that defines what is to be processed by the lens function. The second is a *join*  $\bowtie$  that specifies how the result of the lens function is to be integrated back into the standard pipeline.

**The Selection  $\sigma$**  The selection captures what is to be affected by a lens. Typically, the user controls the selection by moving the lens directly on top of the visual representation of the data. The selection then corresponds to the content shown underneath the lens.

As illustrated in Figure 4, in principle the selected content can be any information that is available along the visualization pipeline. Defined in screen space, a lens can be used to directly select a set of pixels from the visualization view ( $V$ ). By appropriate inverse projection, content from other stages of the visualization pipeline can be selected, be it a set of 2D or 3D geometric primitives at  $VA$ , a group of data entities at  $DT$ , a range of values at  $DS$ , or a combination thereof.

For example, the fish-eye distortion lens from the introduction (see Figure 2(c)) performs the selection on dot positions at  $VA$ . To this end, the lens is back-projected from the screen space ( $V$ ) to the model space ( $VA$ ), in which the geometry and graphical properties of the visualization are defined. Further inverse projection to the data space ( $DT$  or  $DS$ ) enables selection at the level of data entities or data values. For example, with the *ChronoLenses* [ZCPB11] from Figure 5(a), the user basically selects an interval on a time scale. The *Local Edge Lens* [TAvHS06] from Figure 7(a) selects data entities, more specifically, the graph edges that are connected to nodes that are located in the lens.

To drive the selection of the content to be affected by a lens it makes sense to assign unique identifiers to data items and maintain them throughout the visualization process. For large data it can be helpful to utilize half-spaces [TFS08] and efficient spatial data structures [Sam06].

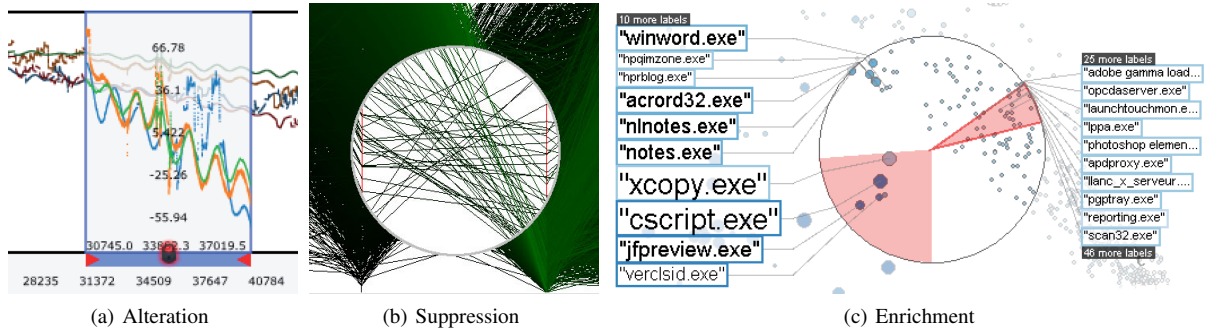
Typically, the selection will be a proper subset that is significantly smaller than the original data. This allows a lens to perform calculations that would take too long for the entire dataset or would not be possible at all. To maintain operability, some lenses install mechanisms that restrict the selection. For example, the *extended excentric labeling* lens [BRL09] from Figure 5(c) adjusts its extent automatically to keep the selection at a manageable size.

Another interesting case are lenses whose modifications of the visualization potentially influence or feedback to the selection. This can happen when the lens function modifies information that is at the same time the basis for the selection. An example is the *Layout Lens* [TAS09] from Figure 6(a) whose selection is based on positions of nodes and that also modifies these positions. To work properly, the lens must not recursively update the selection while nodes are repositioned on the display.

**The Lens Function  $\lambda$**  The lens function defines how the visualization is modified. It creates the intended lens effect. The selection  $\sigma$  is the input to the lens function. The lens function further depends on parameters that control the lens effect. A magnification lens, for example, may expose the magnification factor as a parameter. A filtering lens may be parameterized by a threshold to control the amount of data to be filtered out and by an alpha value used for dimming.

To achieve the lens effect, the lens function processes the selection through the stages of the visualization transformation. Depending on the selection, it can be fully sufficient to process only some stages. For example, when the selection





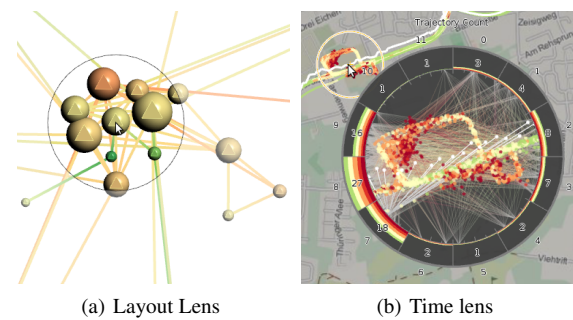
**Figure 5:** Basic lens functions. (a) ChronoLenses [ZCPB11] alter existing content; (b) the Sampling Lens [ED06b] suppresses content; (c) the extended excentric labeling lens [BRL09] enriches with new content.

is defined on pixels, the lens function usually manipulates these pixels exclusively at the view stage  $V$ . A selection from the data source  $DS$  opens up the possibility to process the selected values through all stages to create an entirely different visual representation.

The output generated by the lens function will typically lead to an alternative or modified visualization of the data. From a conceptual point of view, a lens function can *alter* existing content, *suppress* irrelevant content, or *enrich* with new content, as well as perform combinations thereof. Figure 5 illustrates the different options. For example, *ChronoLenses* [ZCPB11] transform time series data on-the-fly, that is, they alter existing content. The *Sampling Lens* [ED06b] suppresses data items to de-clutter the visualization underneath the lens. The *extended excentric labeling* [BRL09] is an example for a lens that enriches a visualization, in this case with textual labels.

**The Join**  $\bowtie$  Finally, the result obtained via the lens function has to be joined with the base visualization. Conceptually, this can be done at any stage of the pipeline. Most lenses follow the metaphor of conventional lenses and simply replace the pixels in the lens interior with the result generated by the lens function. Examples are the *ChronoLenses* [ZCPB11] and the *Sampling Lens* [ED06b] in Figures 5(a) and 5(b). For many more lenses, the visual effect manifests exclusively in the lens interior.

When the join is realized at early stages of the visualization pipeline, the visual effect is usually less confined. For example, the *Layout Lens* [TAS09] in Figure 6(a) adjusts the positions of a subset of graph nodes to create a local neighborhood overview. Relocating nodes implies that their incident edges take different routes as well, which in turn introduces (limited) visual change into the base visualization beyond the lens. Under a very loose interpretation of the lens metaphor, the result of the lens function can even be shown separately. The *time lens* [TSAA12] depicted in Figure 6(b) is such a border-line example where the lens selection takes

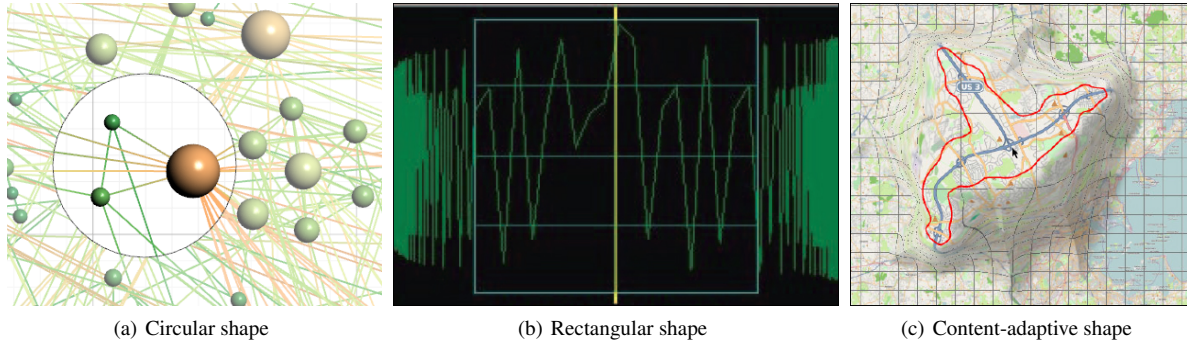


**Figure 6:** Lenses with effects beyond the lens interior. (a) Adjusting layout positions of graph nodes via the layout lens [TAS09] also affects edges in the base visualization. (b) The time lens [TSAA12] shows the lens selection and the generated lens result separately.

place in one part of the display (top-left circle around the mouse cursor) while the generated visual result is shown in another part (center). In such cases, it is important to make sure that the user understands how the lens view relates to the base visualization.

In addition to combining the lens effect with the base visualization, the join is responsible for integrating suitable visual feedback. Showing the lens outline makes clear to the user that a lens is in operation and where the selection takes place. Dimming the base visualization as we have already seen in Figure 2 is an additional visual cue that guides the user's attention to the effect shown within the lens.

An interesting question is how the join is carried out technically. A plausible solution is a three-step procedure. First, one renders the base visualization, optionally sparing the interior of the lens. Second, the result of the lens function is fetched to the lens interior, optionally blending with the base visualization. Third, additional visual feedback is rendered, including the lens geometry and optional user control ele-



**Figure 7:** Lenses with different shapes. (a) Circular Local Edge Lens [TAvHS06]; (b) Rectangular SignalLens [Kin10]; (c) Content-adaptive JellyLens [PPCP12].

ments. On modern GPUs, these operations can be accelerated via multiple render buffers, blending masks, and individual shader programs for the different steps.

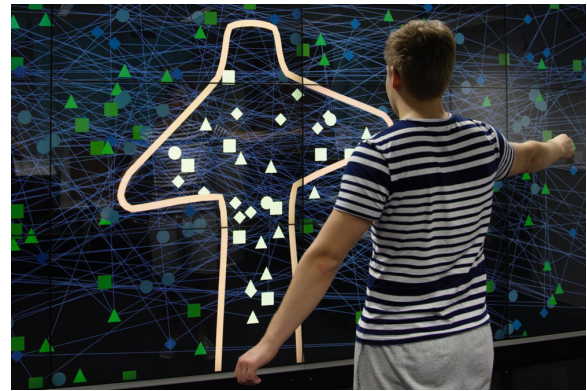
However, this is not a universal solution. Instead the specific implementation details largely depend on the lens function and the desired effect. For example, the *Layout Lens* [TAS09] from Figure 6(a) is rather easy to implement as it only adjusts node positions at the stage of visual abstractions (VA). So the join is merely to override the default positions in the standard pipeline with the adjusted positions as computed by the lens. On the other hand, the join can be rather complicated, for example when considering different layers of 3D graphical information that need to be merged in correct depth order [KMS09].

In summary, selection  $\sigma$ , lens function  $\lambda$ , and join  $\bowtie$  are the key components of our conceptual model of lenses for visualization. Key properties of lenses will be described in the next section.

## 2.2. Lens Properties

From a user perspective, properties that are related to the lens geometry are most evident. The geometric properties not only define a lens' appearance, but also determine the selection, that is, the part of the data where the lens takes effect. The key geometric properties to look at are: shape, position and size, as well as orientation.

**Shape** The most prominent property of a lens is its shape. In theory, there are no restrictions regarding the shape of a lens. It is usually chosen to meet the requirements of the application and to go hand in hand with the lens function. Following the classic prototype of real-world lenses, many virtual lenses are of circular shape. An example is the *Local Edge Lens* [TAvHS06] as shown in Figure 7(a). Yet, in a world of rectangular windows on computer screens, it is not surprising to find lenses of rectangular shape, such as the *SignalLens* [Kin10] in Figure 7(b).



**Figure 8:** BodyLenses [KRMD15] can automatically adapt their shape by sensing a user's posture.

An interesting alternative are lenses that are capable of adapting their shape automatically based on characteristics of the data. Such self-adapting lenses are particularly useful in cases where the lens effect needs to be confined to geometrically complicated features in the visualization. Examples are the *smart lenses* [TFS08], which adjust themselves to shapes of geographic regions, or the *JellyLens* [PPCP12], which can morph itself according to arbitrary geometric features. A JellyLens is depicted in Figure 7(c).

Self-adjustments are not restricted to properties of the data. *BodyLenses* [KRMD15] can adapt themselves to the user. By sensing users in front of a wall display, BodyLenses can dynamically modify their shape so as to resemble the users' posture. Figure 8 illustrates this mechanism. User-adaptive shapes can be useful, for example, to separate private and public areas in collaborative working environments.

While two-dimensional shapes are prevalent, there are also lenses with 3D shapes. They are typically used for the visualization of 3D data or 3D graphics in general. Examples

are lenses for flow visualization [FG98], for magnification of volume data [LHJ01], lenses for 3D scenes [CCF97, RH04], or bendable virtual lenses [LGB07].

**Position and Size** Virtually all lens techniques are parameterizable in terms of position and size. These properties are key to the flexibility offered by interactive lenses. Adjusting the position of a lens will set its focus on a different part of the data, whereas scaling a lens up and down controls the extent of the lens effect. Taken together, moving and resizing enables users to properly attune a lens to the data features they are interested in. Moreover, controlling the size of a lens is a useful means to maintain interactivity and comprehensibility at all times. This concerns the resources required for computing the lens effect, but also the cognitive effort required to make sense of the lens effect in relation to the base visualization.

A large part of the interactivity attributed to lens techniques pertains to direct manipulation [Shn83] of lens position and size. Complementary to interactive adjustment are methods to set position and size of a lens automatically. For example, to guide users to interesting events in the data, one can automatically position a *Table Lens* [RC94] according to those data tuples for which events have been detected [Tom11]. The *RouteLens* [AACP14], while being moved, automatically adjusts its position according to a route on a map to make the route easier to follow. As already mentioned, an example of a lens that automatically adjusts its size to cope with expected costs is the *extended excentric labeling* lens [BRL09]. This lens reduces its size when being moved into areas where too many items are present. Reducing the lens size limits the number of items to be labeled, which in turn helps to maintain interactivity.

**Orientation** A less apparent property of a lens is its orientation. In classic 2D applications, lens orientation is indeed less relevant and therefore often neglected. In fact, for circular 2D lenses, orientation does not even make sense. Yet, orientation can be useful for fine-tuning a lens or is even needed to define the selection at all. As illustrated in Figure 9(a), the *PaperLens* [SSD09] relies on 3D orientation to define a slice through a virtual volume in the space above a tabletop display. For another example consider the elliptical *FlowLens* [GNBP11] in Figure 9(b) and how it is oriented to better match the underlying visualization.

Orientation becomes more important in settings where multiple users look at a common display from different angles, as it is the case for collaborative work at a tabletop display [VTS\*09]. An example is a user who is handing over a lens to a co-worker at the opposite side of the table, which for certain involves re-orienting the lens.

In summary, we see that lenses exhibit several key geometric properties that can be interactively or automatically adjusted. In addition to what we described here, there can be further parameters, for example, a depth range for 3D lenses

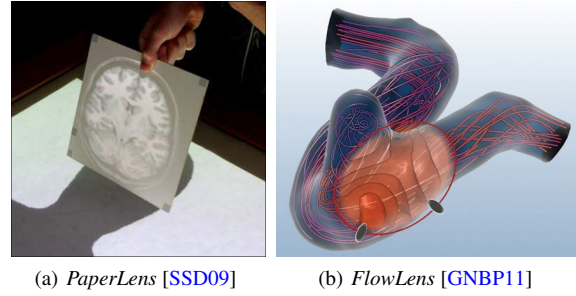


Figure 9: Differently oriented lenses.

or a drop-off interval for fuzzy lenses. Usually, such parameters are interactively adjustable via standard controls or dedicated lens widgets as illustrated in Section 6. An example of a lens with automatic parameter adjustment is the *Sampling Lens* [ED06b]. This lens can tune its sampling rate parameter to the density of the underlying data plot.

### 2.3. Discussion

So far, we have identified conceptual components (selection  $\sigma$ , lens function  $\lambda$ , and join  $\bowtie$ ) and properties (shape, size and position, orientation) of lenses. Next, we discuss implications for design and implementation.

**Stages of Selection and Join** We have indicated that lenses can relate to the stages of the visualization pipeline in different ways. The selection and join may gather input and return output at any stage. The lens function may generate the lens effect by implementing any subset of the pipeline. There are two noteworthy observations. First, if the selection takes place at the early stages, the lens function has more options for creating a richer lens effect. For example, the *extended excentric labeling* [BRL09] selects data items, which allows the lens function to generate textual labels and place them appropriately. The second observation relates to the dependency of a lens effect's extent on the stage at which the join takes place. Effects constrained to the lens interior are typically the result of joining at the pixel stage as for the *Sampling Lens* [ED06b]. Joining at earlier stages, as for the *Layout Lens* [TAS09], can have side effects on the base visualization.

**Lens Interior, Exterior, and Border** Another aspect worth discussing is that a lens, described by its geometric properties, naturally partitions the display space into three regions: inside, outside, and border (see Figure 3). We already talked about the lens interior, where typically the lens effect is shown, and about the lens exterior, where usually the base visualization remains relatively unchanged and where additional visual cues may be incorporated. The border of a lens is equally important. Visually, the border clearly communicates the separation of lens interior and exterior, thus helping



the user in defining the selection precisely. We will later see in Sections 4 and 6 that the lens border also plays an important role in terms of interaction. The lens border itself can serve as a handle to adjust the lens and it is a suitable place to attach additional user interface controls enabling the direct manipulation of lens parameters, as for example for the *FlowLens* [GNBP11] in Figure 9(b).

**Combining Lenses** Until now, we have considered altering a visualization with a single lens. Yet, our conceptual model in Figure 4 also indicates that it is possible to use multiple lenses to generate a desired effect. A combination of lenses can be realized by iteratively attaching multiple lens pipelines to a standard pipeline, or by combining lenses recursively. In the latter case, a lens pipeline is attached to another lens pipeline. The *Composite Lens* [TAS09] is an example of an iterative combination of three lenses, the *Local Edge Lens* from Figure 7(a), the *Layout Lens* from Figure 6(a), and the fish-eye lens from Figure 2(c). The Composite Lens allows users to switch on and off individual lenses as necessary. Yet, it is based on a hard-wired template tailored to the involved lens techniques. Flexibly combining arbitrary lens techniques is considerably more difficult [Fox98]. It requires proper execution of multiple interdependent selections and elaborate means for blending lens results in a conflict-free way [TFS08]. All this is not easily accomplished in visualization scenarios and remains a challenge to be addressed in the future as discussed in Section 7.

In the next section, we broaden our view and survey actual lens techniques for various types of data and user tasks.

### 3. Data, Tasks, and Lenses

By considering data and tasks, we shift our focus to aspects that are more of interest to the users of visualization tools. Such users may ask: I'm working with this and that type of data, is there a lens for the task I have to accomplish with my data? The following paragraphs aim to provide an answer to this question. As a first part of the answer, we will look at lenses for different types of data. For the second part of the answer, we will review lenses for different tasks.

#### 3.1. Lenses for Specific Data

There are general purpose lenses and there are lenses that are specifically designed for a particular type of data. Lenses that operate on pixels or geometric primitives are usually oblivious to the specific data type of the underlying visualization. For example, magnification lenses (e.g., [Kea98, CM01, CLP04, FS05, PPCP12]) are techniques that are universally applicable across many, if not all types of data.

In this section, we review lenses that are tailored to the characteristics of specific types of data. Shneiderman's [Shn96] taxonomy collects several data types being relevant in visualization. For our survey of data-specific

lenses, we use a slightly different set of data types, which we think to be a bit more descriptive:

- Temporal data
- Geospatial data
- Flow data
- Volume data
- Multivariate data
- Graph data
- Text and document data

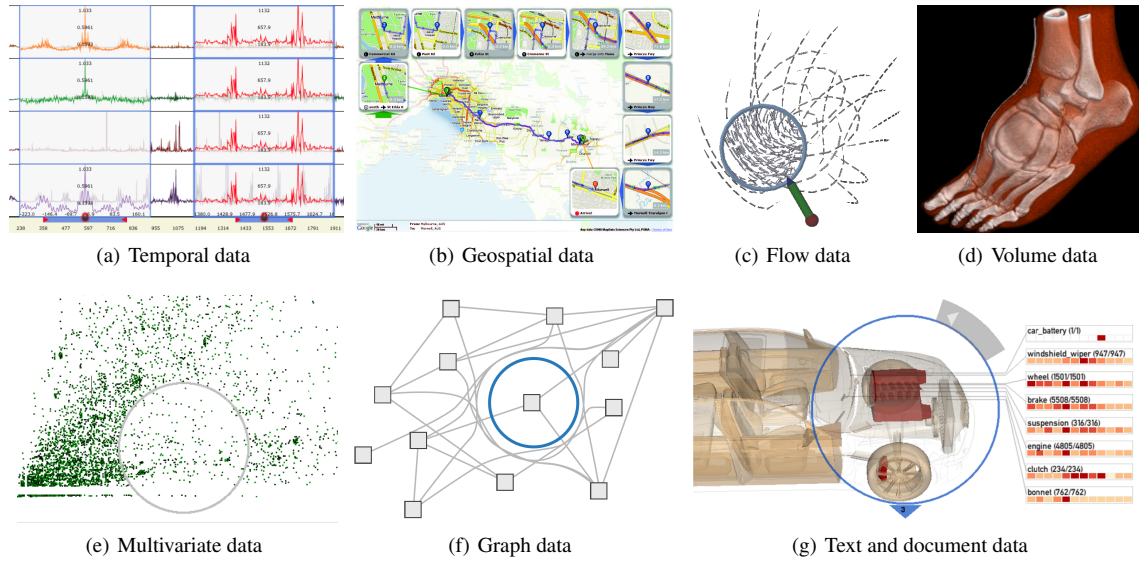
The purpose of the following paragraphs is to illustrate the diversity of existing lens techniques. To this end, we describe one exemplary lens for each data type very briefly. A more comprehensive overview of existing lenses for different types of data is provided in Table 1 on page 12.

**Temporal Data** Data that have references to time are commonplace. Time series allow us to understand the development of past events and predict the future outcome of ongoing phenomena. Analyzing time series often involves normalization, elimination of seasonal effects, and other temporal transformations. *ChronoLenses* [ZCPB11] enable users to carry out such operations on-the-fly for selected time intervals. The transformed part of the time series is overlaid with the original version to maintain the context and facilitate comparison. As illustrated in Figure 10(a), *ChronoLenses* can be applied to multiple stacked time series simultaneously.

**Geospatial Data** Data about the world around us often hold references to the geospatial domain, describing where certain phenomena or features have been observed. Maps are frequently used as a basis for visualizing the spatial frame of reference, which in turn serves to depict the actual geospatial data. As maps are omnipresent in our everyday lives, it is not surprising to find a lens developed for a routine problem: the depiction of driving directions. *Detail Lenses* [KCJ\*10] aim to make the visual representation of driving directions on maps easier to follow. To this end, relevant points of interest (POI) are defined along a route. For each POI, a lens with a detailed map of the POI is arranged around a global overview map as illustrated in Figure 10(b). The arrangement is computed so as to make it easy for the user to associate the lenses with the POIs along the route.

**Flow Data** 2D or 3D vectors are the basic components of flow data. In addition to the raw vectors, derived scalar attributes, higher-level features (e.g., vortices), and flow topology are relevant aspects of the data. Fuhrmann and Gröller [FG98] are among the first to propose lens techniques specifically for flow data. They use a planar 3D lens polygon (see Figure 10(c)) or a volumetric 3D lens box to implicitly or explicitly define a 3D spatial selection in the flow for which more details are to be shown. The low-resolution base visualization is replaced by filling the lens interior with more and thinner streamlines that also have an increased number of control points.





**Figure 10:** Lenses for different data types. (a) ChronoLenses [ZCPB11] for temporal data; (b) Detail Lenses [KCJ\*10] for geospatial data; (c) Magic Volume Lens [WZMK05] for volume data; (d) Magic lens for flow data [FG98]; (e) Sampling Lens [EBD05, ED06b, ED06a] for multidimensional data; (f) PushLens [SNDC10] for graph data; (g) Lens widget for querying document data [CC13].

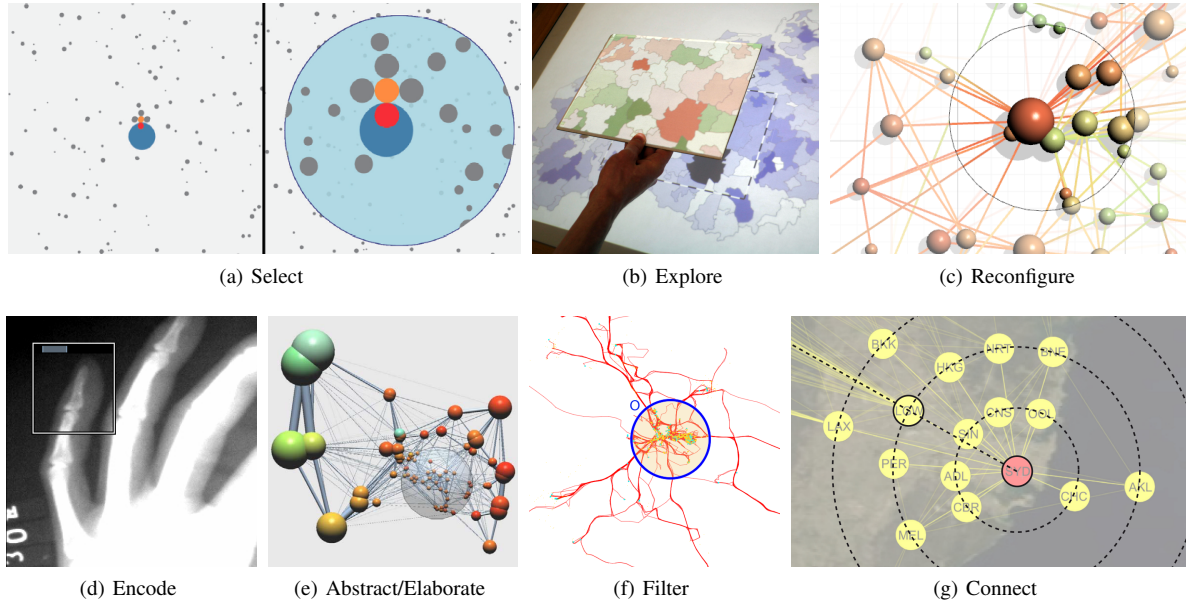
**Volume Data** Volume data are scalar data values on a three dimensional grid. Hence, lenses for volume visualization face the typical challenges related to spatial selection and occlusion in densely packed 3D visual representations. The *Magic Volume Lens* [WZMK05] addresses these challenges. In fact, the authors propose different lenses to magnify volumetric features, while compressing the context without clipping it entirely. Figure 10(d) shows a *Magic Volume Lens* applied to magnify a section of a human foot. The selection can be made interactively or can be adapted to data features automatically. The lens effect is directly embedded into the direct volume rendering through a multi-pass rendering strategy.

**Multivariate Data** General multidimensional, multivariate data typically span a frame of reference across multiple abstract dimensions. Along these dimensions multiple quantitative or qualitative variables are measured. For such data, dealing with data size is usually a challenging problem. The *Sampling Lens* [EBD05, ED06b, ED06a] addresses the problem of clutter in scatter plots and parallel coordinates. Through sampling, the lens function generates a more sparsely populated alternative view on the data. The selection made with the lens is inversely projected into the data space in order to estimate a suitable sampling rate for the alternative visual representation to be shown inside the lens. The clutter-reduced visual representation of the sampled data is rendered and clipped at the lens boundaries as shown in Figure 10(e).

**Graph Data** A graph is a universal model for describing relations among entities. Although sophisticated graph layout algorithms exist, visual representations of graphs might still suffer from ambiguities. The *EdgeLens* [WCG03] (and later the *PushLens* [SNDC10]) address this problem by reducing the number of edges being visible inside the lens. The selection singles out edges that intersect with the lens geometry, but whose source and target nodes are not inside the lens. Then the lens interior is cleared of the selected edges by ‘bending’ them around the lens. Figure 10(f) illustrates edges that are routed around the *PushLens*.

**Text and Document Data** Text and document collections are rich sources of information. Analysis of text typically focuses on important topics or themes and querying relevant keywords. A particularly interesting application of a lens for exploring text databases is the approach by Chang and Collins [CC13]. What makes their lens special is the fact that they combine spatial and abstract semantics in a single view. As illustrated in Figure 10(g), the lens selection operates on a 3D model of a car capturing the spatial semantics. Associated with the spatial selection are keywords, which capture the abstract semantics. Keywords corresponding to the selection are visualized as heatmap charts around the lens and an additional panel provides on-demand access to the underlying text documents.

In the previous paragraphs, we have briefly reviewed examples of lenses for different types of data. In the next section, we will focus on lenses for specific tasks.



**Figure 11:** Lenses for different tasks. (a) The Bubble Lens [MW14] assists in selection tasks; (b) Tangible Views [STSD10] used to explore data; (c) the Layout Lens [TAS09] reconfigures nodes in a graph layout; (d) the Color Lens [EDF11] adapts the encoding of colors; (e) Using a lens to automatically expand and collapse nodes of a graph [vHvW04]; (f) TrajectoryLenses [KTW\*13] filter movement trajectories; (g) Bring & Go [MCH\*09] helps in connect tasks.

### 3.2. Lenses for Specific Tasks

Lenses are designed as interactive tools. So it makes sense to review lens techniques in terms of the interaction tasks they do support specifically. In order to structure such a review, we resort to the seven key categories of user intents for interaction as identified by Yi et al. [YKSJ07]:

- Select – Mark something as interesting
- Explore – Show me something else
- Reconfigure – Show me a different arrangement
- Encode – Show me a different representation
- Abstract/Elaborate – Show me more or less detail
- Filter – Show me something conditionally
- Connect – Show me related items

Each intent captures why a user wants to or needs to interact while carrying out exploratory or analytic tasks. Similar to what we have done for different types of data, we will now very briefly describe exemplary lenses for each category to illustrate their utility for the different interaction tasks. For a more complete list of lens techniques and their suitability for the different interaction tasks, we again refer to Table 1 on page 12.

**Select – Mark Something as Interesting** A typical problem in dense visual representations is to pinpoint data items of interest. An example of lens techniques that supports this task is the *Bubble Lens* [MW14]. This lens makes picking

small data items easier by actually enlarging not only the visual space in which visual feedback is shown, but also the motor space in which users interact. The magnification is automatically applied when the *Bubble Lens* is moved towards a region containing many small data items. Figure 11(a) illustrates the *Bubble Lens* entering such a region (left) and the ensuing magnified view (right).

**Explore – Show Me Something Else** Exploration relates to undirected search, the user is interested in seeking out something new. The *tangible views* for information visualization [STSD10] demonstrate a variety of exploratory lens methods. An example is the exploration of space-time cubes via sliding a tangible view through a virtual space-time continuum above a tabletop display. This enable the user to focus on different points in time and different places in space. Figure 11(b) illustrates such a tangible lens above a map.

**Reconfigure – Show Me a Different Arrangement** The spatial arrangement of data items on the screen is key to comprehending the visualized information. Looking at different arrangements helps to gain a better understanding of the data. A lens that temporarily rearranges the visual representation is the *Layout Lens* [TAS09] presented in Figure 11(c). The idea of this lens is to create a local overview of the neighborhood of the nodes inside the lens. For this purpose, the node layout is reconfigured locally to bring all relevant neighbors to a position within the lens interior.

**Encode – Show Me a Different Representation** The visual mapping decides about the expressiveness of a data visualization. Yet a global visual encoding of the data as a whole can make it difficult to spot details in local regions. The *Color Lens* [EDF11], for example, adapts the color coding according to the local conditions underneath the lens. Multiple lenses can be used to create adapted visual representations for different parts of the data, or alternatively, the global view can be replaced by a selected adapted representation temporarily. The latter case is illustrated in Figure 11(d) where the global contrast and brightness have been adjusted according to the image content inside the lens.

**Abstract/Elaborate – Show Me More or Less Detail** This rather general task is supported in many different ways. Various general purpose lenses have been proposed that show more detail through magnification. The lens technique by van Ham and van Wijk [vHvW04] provides access to more or less details about clusters in a hierarchical graph. Nodes approached with the lens are automatically expanded, whereas nodes that are no longer in the focus of the lens are automatically collapsed. An application of this lens to a small world graph is shown in Figure 11(e).

**Filter – Show Me Something Conditionally** Filtering according to specific criteria, while still maintaining a global overview is an important task when exploring data. In some cases, filtering even acts as an enabling technique for the visualization. For example, the filtering capabilities of *TrajectoryLenses* [KTW\*13] enable the visual exploration of massive movement data. Filtering can be done according to origins and destinations of movements, regions passed by movements, and time intervals spanned by movements. Figure 11(f) shows a lens with an origin (O) filter.

**Connect – Show Me Related Items** Once the user has focused attention on data items of interest, the exploration usually continues with related or similar data. The *Bring & Go* approach [MCH\*09] is a lens-like technique that supports navigation to related items in graphs. All nodes that are related (i.e., connected) to a focus node are brought close to the focus. Clicking on a connected node triggers an animated trip to that node. Figure 11(g) depicts an example where Sydney (SYD) has been selected as the focus. Airports directly connected to SYD are arranged on concentric circles depending on their relative distance.

This section illustrated that lenses are broadly applicable to various data and tasks. As we could only briefly describe a few selected lens techniques, the reader is referred to Table 1, which categorizes 50 original publications according to data type and user task. Still more references and additional brief captions are collected in the appendix.

The next section will shift the focus from data and tasks to how users actually work with lenses.

## 4. Interaction and Display

Interactive lenses in visualization take advantage of various interaction modalities and show their visual feedback on different kinds of displays. Interactive operations that have to be considered include creation and deletion of lenses, manipulations of the lens geometry (recall the properties discussed in Section 2.2), as well as more complex operations, such as parameterizing the lens function or combining multiple lenses. However, these interactive operations are rarely discussed in detail, because usually it is the lens function that is in the focus.

The goal of this part of our survey is to illustrate the wide variety of interactive visual environments that serve as a technical basis for interactive lens techniques. Illustrating examples of different display settings and interaction modalities are given in Figure 12. In the next sections, we will discuss interaction with lenses in the context of a variety of styles and modalities. These include:

- Mouse and keyboard interaction
- Touch and multi-touch interaction
- Tangible interaction
- Tangible views and spatial interaction
- Gaze-based interaction and head tracking

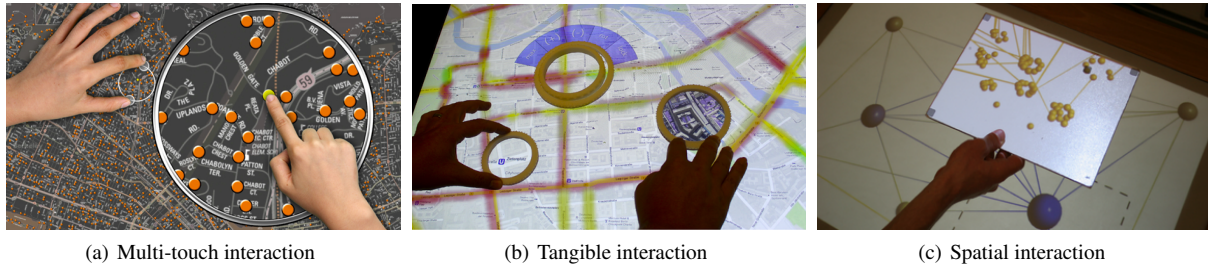
We start with looking at traditional visualization environments, which typically consist of a single display and mouse and keyboard input devices. However, such environments quickly reach their limits in the face of complex analysis tasks performed on big data. The single display limits the amount of information that can be shown at a time and mouse and keyboard interaction is limited to a single user.

New visualization environments address these limitations in different ways. Interactive tabletop displays [IIIH\*13] and multi-display environments [EBZ\*12] provide more display space for the visualization and enable multiple users to work collaboratively. Specifically because of their local effect, lenses in combination with large interactive surfaces or displays can become powerful tools for collaboration. A good example is the work by Waldner et al. [WKS07]. In their setting several users can gain an alternative view on the data through optical lenses by utilizing polarized light on large-scale projector-based displays. As more than one person can work with the visual representation of the data, joint data exploration and analysis becomes less tedious. Yet, there are also additional challenges that need to be addressed. On tabletops, the orientation problem arises when multiple users stand on different sides of the table looking at lenses and visualizations. On large high-resolution displays, covering large distances causes considerable difficulties. Context-dependent interaction techniques are required to enable multiple users to make the best of interactive lenses in such new visualization environments.

Techniques	Data						Tasks							
	Temporal	Geospatial	Flow	Volume	Multivariate	Graph	Document	Select	Explore	Reconfigure	Encode	Abstract & Elaborate	Filter	Connect
[SB92,SB94] Fisheye Views		•				•						•		
[RM93] Document Lens							•					•		
[CMS94] MagicSphere				•				•			•	•	•	
[RC94] Table Lens					•							•		
[VCWP96] 3D Magic Lenses				•								•	•	
* [FG98] Lenses for Flow Visualization			•								•	•		
[FP99] Excentric Labeling					•				•			•		•
[SHER99] Interactive 3D Lens			•	•	•		•				•	•	•	
[LHJ01] Volume Magnification Lens				•								•		
[SFR01] Time Lens	•								•					
[BCPS02] Fuzzy Lens, Base-Pair Lens, Ring Lens						•				•	•	•		
[MTHG03] 3D Flow Lens			•								•	•		
* [WCG03] EdgeLens						•				•				
* [vHvW04] Graph Abstraction Lens						•						•		
[RHS05] Magic Lenses in Geo-Environments		•									•		•	
* [EBD05,ED06b,ED06a] Sampling Lens					•								•	
[RLE05] Temporal Magic Lens	•								•					
* [WZMK05] The Magic Volume Lens				•								•		
[TAvHS06] Local Edge Lens						•							•	
[KSW06] ClearView				•								•	•	
[TGBD08] 3D Generalization Lenses		•										•		
* [TAS09] Layout Lens						•				•				•
[BRL09] Enhanced Excentric Labeling					•				•			•		•
* [MCH*09] Bring &Go						•				•				•
[ACP10] High-Precision Magnification Lenses	•	•	•	•	•	•	•	•				•		
[JDK10] Network Lens						•					•			
* [KCJ*10] Detail Lenses for Routes		•										•		•
[Kin10] SignalLens	•											•		
* [SNDCl0] PushLens						•				•				
* [STSD10] Tangible Views		•			•	•		•	•	•	•	•	•	•
* [EDF11] Color Lens	•	•	•	•	•	•					•			
[GNBP11] FlowLens			•						•		•		•	
[HLTE11] SemLens					•						•		•	
[HTE11] MoleView		•			•	•				•				
[LWG11] Facet Lens							•				•		•	
[PBKE11] EdgeAnalyser					•	•					•	•		
[ZCB11] MagicAnalytics Lens	•													•
* [ZCPB11] ChronoLenses	•												•	•
[TSAA12] Time Lens		•							•		•			•
[PPCP12] JellyLens		•										•		
* [KTW*13] TrajectoryLenses		•											•	
[PPCP13] Gimlenses				•								•	•	
[UvK13] Magic Lenses for Hypergraphs						•							•	
* [CC13] Lens for Querying Documents							•						•	•
[AACP14] RouteLens		•										•		
[BHR14] PhysicLenses		•										•		
* [MW14] Bubble Lens					•			•			•			
[DMC15] VectorLens	•				•			•					•	
[KRD14] Multi-touch graph Lenses						•				•			•	•
[DSA15] 3DArcLens		•				•				•			•	

**Table 1:** Lens techniques in the context of visualization categorized according to data types and tasks. Entries are sorted in chronological order. Techniques for which a brief description is available in Section 3 are marked with \*.





**Figure 12:** Examples of different interaction modalities for lenses. (a) Bimanual touch interaction in FingerGlass [KAP11]; (b) graspable interaction through TangibleRings [EWCP13]; (c) interaction above the tabletop with Tangible Views [STSD10].

#### 4.1. Mouse and Keyboard Interaction

Mouse and keyboard interaction has been the most common interaction modality of the last decades. Therefore, it is no surprise that it is also the most prominent in the reviewed research on lenses. The typical environment of mouse and keyboard interaction is a one-to-few display setup for a single user. The precision of the cursor movement is the major advantage of mouse input. Additional possibilities of the mouse are given through mouse buttons and the mouse wheel, which can be used for discrete or continuous input, respectively. Keyboard input, on the other hand, is suitable for mode switches or step-wise navigation.

The precise direct manipulation of the mouse is especially useful when specifying the region of interest and hence repositioning of the lens. This technique is used in many of the examined research works, for instance in [BCPS02, ED06a, PPCP12, TAvHS06, WCG03]. As the lens position is coupled with the mouse cursor, fast repositioning becomes possible. However, when a magnification lens is used, this fast positioning at context scale can hinder target acquisition in the focus area and make pixel-precise positioning impossible [ACP10].

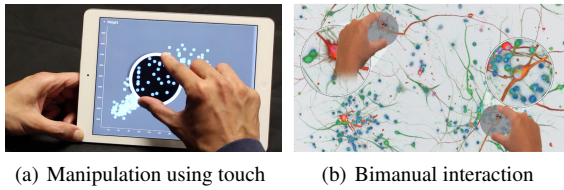
Appert et al. [ACP10] introduce different interaction techniques to improve on the target selection when using magnification lenses. First, they present mode switching between focus and context speed by using a keyboard key. Second, they provide a continuous mapping between precision of lens movement depending on cursor speed. The third interaction technique introduces a ring as the lens border where the lens' inner region is navigated at focus scale while the outer region is navigated at context scale. In the experiments, all techniques performed better than regular lens positioning and the ring technique performed best in experiments for small targets and high magnification factors without needing the additional keyboard mode switch. For the specific *Route-Lens*, Alvina et al. [AACPI14] use the knowledge of the underlying data, paths and streets, to predict the lens movement and counteract overshooting due to the magnification of the lens.

To incorporate the adjustment of parameters other than position, mouse buttons are used to toggle a specific parameter or state [BRL09, HTE11]. Additionally, the mouse wheel can be helpful when changing continuous values, such as attribute ranges [HTE11]. However, as more and more parameters have to be adjusted for complex lens functions in visualization, graphical user interfaces are necessary. Possible controls and widgets include toolbars, combo boxes, or sliders. The mouse is then used to adjust these parameters in either global or context menus. Some examples can be found in Jusufi et al.'s work where a dialog box is used to create and edit the *Network Lens* [JDK10]. The *Sampling Lens* [EBD05] is another example, as a slider is used for lens diameter adjustment.

In three-dimensional visualizations, adjusting the lens geometry in relation to the data becomes difficult. For CAD models, Pindat et al. [PPCP13] use mouse interaction for positioning the lens with the help of automatic orientation constraints and virtually drill into the visualization using the mouse wheel. For their lenses for 3D flow visualization, Fuhrmann and Gröller [FG98] suggest a 3D mouse or tracked pen to reconfigure the six degrees of freedom. The *Document Lens* [RM93] is controlled with mouse interaction for the x-y-plane positioning and with keyboard keys for movement within the z-plane.

#### 4.2. Touch and Multi-Touch Interaction

In the past decade, touch devices have become increasingly commonplace. Especially noticeable is the upsurge of touch-enabled personal devices, such as smartphones and tablets, but also interactive displays for desk work, such as touch-screen monitors, became accessible. Additionally, touch-enabled devices can be enhanced with digital-pen recognition. Such devices allow for direct interaction in context and natural interaction with virtual elements. In a sense, direct manipulation becomes truly direct thanks to the unification of input and output space. This is generally beneficial for visualization, as illustrated by the natural interaction techniques for stacked graphs by Baur et al. [BLC12].



**Figure 13:** Advantages of touch: (a) Direct interaction with data points [SS14] and (b) bimanual manipulations with focus on multiple regions of interest [BHR14].

In terms of lens interaction, especially the adjustment of the geometric properties discussed in Section 2.2 becomes natural and does not require additional widgets. Manipulations (translation, scaling, or rotation) can be accomplished by direct interaction through multi-touch gestures and no menu widgets are necessary for this part. Hence, the user can fully concentrate on the vital aspects – the lens function and its effect on the visualized data. However, one limitation of touch interaction is the *fat-finger problem* [VTS\*09], which describes the ambiguity in position that arises from the imprint of the soft fingertip together with the occlusion of the target object by the finger. Hence, pixel-precise positioning of touch requires additional interaction concepts. Different solutions have been proposed to overcome this problem (e.g. in [BWB06]) by using an offset between finger and cursor, interfering with the directness of the interaction, or use of digital pens for smaller contact input.

In terms of lenses, Schmidt et al. [SNDC10] address interaction with node-link-diagrams by designing touch gestures for edge manipulation and incorporate the creation and manipulation of their *PushLens* through multi-touch interaction. The *PushLens* can be created by using three touches close in space and time and can then be repositioned and resized by touch dragging and pinching on the border. Rzeszotarski and Kittur [RK14] also use multi-touch when positioning their physics-based lens for highlighting elements. Discussing the advantages of multi-touch interaction for scatterplot visualizations, Sadana et al. [SS14] present a zoom lens. This lens is created using a pinch gesture on the data, see Figure 13(a), and allows manipulation of the zoom level as one parameter of the lens function.

As opposed to mouse-based interaction which is always single cursor and one-handed, natural interactions may allow usage of two-handed, multi-touch interactions. Bimanual interaction concepts for lenses are employed in the *FingerGlass* approach [KAP11], which is shown in Figure 12(a). In *FingerGlass*, the selected region of interest is separated from the output view. Both can be interacted with through touch. This technique specifically supports precise object selection and translation. While one hand manipulates the region of interest, elements within the magnified view of the region can be manipulated by the second hand. Butscher et

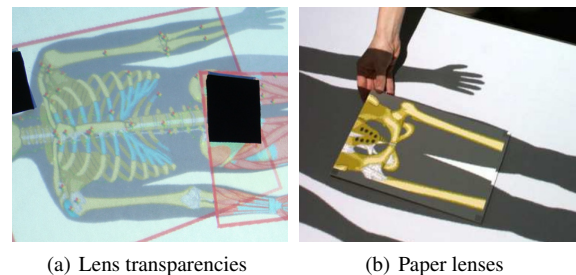
al. [BHR14] use this technique to introduce *PhysicLenses*. A long two-finger tap on the surface initializes the selection for a new, separate lens. *PhysicLenses* apply multiple lenses to the same data view and hence allow comparison of multiple foci, as illustrated in Figure 13(b). As the lens always remains the same size, the distance between the fingers changes the amount of magnification within the lens. To our knowledge, multi-touch interaction techniques for the further setting of parameters related to the lens function have not been examined using touch interaction.

### 4.3. Tangible Interaction

Ishii and Ullmer [IU97] coined the term tangible interaction as early as 1997. The idea was to use the affordances, tangibility, and manipulability of physical objects for an improved interaction with virtual data. Typically, tangibles are used on interactive tabletops to facilitate tracking possibilities and use the space around the tangible for context visualizations. Because of their properties, tangibles can be moved quickly within the reachable area, often even without visual focus, for positioning and change in orientation in a seamless and fluent way.

Kim and Elmqvist present *embodied lenses* [KE12], thin transparencies, that function as additional layers on data similar to *PaperLens* [SSD09]. These tangibles can use physical interactions, such as grasping, placing and moving for creation and manipulation of a lens. Combinations of lenses with different functions is also possible as multiple tangibles can be placed on top of each other. Applications include the exploration of layered data, for example data on the human skeleton and nerve system in medical imaging as shown in Figure 14(a), as well as multidimensional data visualizations, and tangible map interactions.

Ebert et al. [EWCP13] introduce lenses in the form of *TangibleRings*, as illustrated in Figure 12(b). These ring-like tangible objects have two major advantages: they do not occlude or blur the underlying content and they allow for touch



**Figure 14:** Tangible interaction with tracked lenses on interactive tabletops: (a) Embodied lenses [KE12] utilize translucent light from the tabletop; (b) PaperLenses [SSD09] uses projection on paper.

interaction within the region of interest. Additionally, rings with different radius can be concentrically positioned so that lens functions can be logically combined. Similarly, transparent tangibles can be used as graspable representation of lenses that can be stacked and touched from above to manipulate the visible content below the tangible [BKFD14].

A typical limitation of tangibles is their usually inflexible form. Every manipulation of the tangible's shape or size requires hardware modification. This can be easy, when using transparencies or paper or may involve more complex modifications for other materials such as acrylic glass. Therefore, they might only be applicable in specific application contexts. However, their tangibility and the use of our natural motor skills can help make precise and intuitive manipulations of the lens.

#### 4.4. Tangible Views and Spatial Interaction

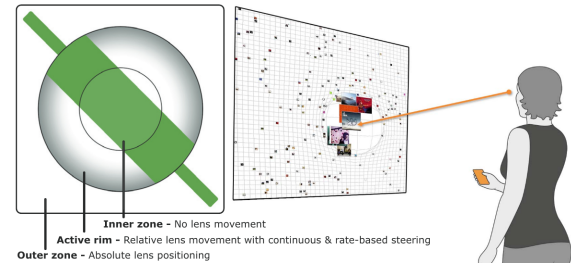
Spindler et al. [SSD09] developed tangible views that are passive paper lenses on which information is projected from above, see Figure 14(b). These tangible lenses combine display and interaction in a single “device” as the display becomes the actual input channel and the interaction is lifted up from the table into the third dimension. The spatial configuration of the paper lens in the x-y-plane above the table determines where the lens effect should be applied. The distance to the table is an additional parameter that can be used for interaction. Later on, these were extended to active tablet lenses [SBWD14] which increases the possible resolution of visualizations on the lens while increasing weight and reducing flexibility in shape.

Interaction concepts such as translation and rotation are possible as much as flipping, tilting and shaking of the lens, which distinguishes this spatial interaction from interaction with tangibles on tabletops. On the tangible view itself, interaction is possible through digital pen or touch. Tangible views have been used for data analysis, such as graph exploration where the distance to the table surface influences the level of abstraction shown in the lens [STSD10]. This use case can be seen in Figure 12(c). Their advantage lies in using the physical coordination and muscle memory of the user to help fast navigation within the interaction space. However, similar to tangibles on tabletops, the limitation of tangible views is their rather fixed size and shape.

Moreover, there are several applications in the field of augmented reality, where smart phones or tablets are used as physical interactive lenses that can be moved freely in space [SSRG11, LBC04, BLT\*12].

#### 4.5. Gaze-Based Interaction and Head Tracking

On large high-resolution displays, controlling lenses with mouse or tangible interaction is infeasible. Gaze-based interaction techniques are a promising alternative [Jac90]. Magnification lenses are often used to make gaze interaction



**Figure 15:** Gaze-based interaction: The gaze will move the magnification lens in outer or border regions only, allowing for examination of the lens content [SD13, SSND11].

more precise, not for the interaction with lenses themselves. Examples for local magnification tools based on gaze dwell time are proposed by Lankford [Lan00] and Ashmore et al. [ADS05]. Yet, dwell times prevent quick execution of commands and hence hinder a fluent workflow, because users need to fixate a point for a certain amount of time.

In further research, a second modality is used for explicit interactions, such as selections. Kumar et al. [KPW07] introduce the concept of *look-press-look-release* to solve this problem: Only when a keyboard key is pressed, the viewed region is enlarged. Stellmach et al. [SSND11] use touch as the second modality for confirmation of selected elements. In these examples, lenses are used to improve gaze as an interaction modality and counteract imprecision. However, interaction with the lens as an interactive tool has also been discussed. Active positioning of these lenses can be cumbersome because of constant eye movement. Stellmach et al. [SD13] developed different regions of movement for the fisheye lens, thereby interacting with the lens through gaze. As described in Figure 15, when looking within the “inner zone” of the lens, no movement is triggered, gaze focus in the “active rim” steers the lens continuously and the “outer zone” helps make fast, absolute positioning.

In the context of graph exploration, Lehmann et al. [LSST11] use head tracking to allow manipulation of a focus+context lens technique by physically moving in front of a large high-resolution display. The lens position is set to the users gaze position approximated by head tracking. Additionally, the distance to the display influences the level of abstraction presented in the lens and thereby directly adjusts a specific function-dependent parameter. Head-tracking is also used by Spindler et al. when designing *Tangible Windows* [SBD12] to provide a volumetric perception on the elements presented on the tangible views when working in 3D information spaces.

To conclude, the majority of lens techniques known in the visualization literature have been developed for mouse and keyboard interaction in desktop environments. Precise positioning of the lens is usually carried out via mouse in-

Trigger type:	Click, drag-out, drag-in, drag-out-and-in, gesture, overlapped tool
Input transparency:	Click-through, click-on
Data direction:	Down, up, both
Data magnitude:	Simple command, ..., database
Application indep.:	No, yes
Application:	Graphics, text, other click-throughs, traditional tools, ...
Operation class:	Select, create, delete, modify, ...
Composition:	Append, prepend, modify
Output transparency:	Opaque, stippled, translucent, transparent
Moves with:	Sheet, application, both, object, cursor
Lens presence:	No, yes
Customization:	None, parameters, editable
Persistence:	Always, temporal mode, appears in context
Complexity:	Single part, ..., many parts

**Table 2:** Taxonomy of see-through tools [BSF\*94].

teraction and lens parameters are to be set in menus and through keyboard shortcuts. There are first promising studies on natural user interfaces for lens interaction, such as tangible lenses and head-tracking for lenses. However, these approaches mostly focus on the manipulation of the position and size of lenses. Few novel interaction techniques have been presented in terms of adjustment of other lens parameters. This is one possible direction for future work on interaction for lenses. Before we present other relevant research questions, we will discuss the presented concepts for one selected lens example.

## 5. Taxonomy of Lenses for Visualization

In the previous sections, we studied lenses from different perspectives. In Section 2, we looked at conceptual aspects. We defined lenses via lens pipelines with selection  $\sigma$ , lens function  $\lambda$ , and join  $\bowtie$ , and described lenses with respect to their properties, including shape, position, size, and orientation as well as parameters of the lens function. In Section 3, we focused on what lenses are good for in visualization settings. The two key criteria were data type and user task. Section 4 was dedicated to aspects of displaying lenses and interacting with lenses in various ways.

It would now be desirable to extract from our collected knowledge a taxonomy of lenses for visualization. This section is an attempt to come up with such a taxonomy. Our taxonomy should provide sufficient reasoning power. It should not only be possible to describe and organize all the lenses

<i>Practical</i>	
Data type:	Temporal, geospatial, flow, volume, multivariate, graph, document, ...
User task:	Select, explore, reconfigure, encode, abstract & elaborate, filter, connect, ...
Display setting:	Small mobile display, desktop, wall display, multi-display environment, ...
Interaction modality:	Mouse+keyboard, (multi-)touch, tangibles, spatial interaction, head-tracking, gaze, ...
<hr/> <i>Conceptual</i>	
Effect class:	Suppress, alter, enrich
Effect extent:	Lens interior, side effects, separate view
Adjustability:	None, interactive, self-adjusting
Selection $\sigma$ stage:	$DS, DT, VA, V$
Join $\bowtie$ stage:	$DS, DT, VA, V$

**Table 3:** Taxonomic extensions for lenses for visualization.

already in existence, but also to reason about new lens-based techniques that have not previously been seen.

An early taxonomy related to magic lenses is presented by Bier et al. [BSF\*94]. Their taxonomy covers 14 axes as listed in Table 2. The taxonomy is useful to study the design space of lenses at a general level.

Here we aim to refine and extend the previous taxonomic efforts according to the insight derived from our survey of lenses for visualization. Our taxonomy should cover *practical* aspects to assist practitioners in finding applicable solutions to real-world problems and *conceptual* aspects to help researchers in studying lenses. Following this thinking, we propose the taxonomic extensions collected in Table 3. The categories and characteristics of our refined and extended schema directly map to the discussions from the previous sections.

As a first step, we revise Bier et al.'s *Application* category to be visualization-specific. To this end, we include two new categories *Data type* and *User task*. The categories *Display setting* and *Interaction modality* have been included to account for the heterogeneous environments in which visualization and lenses are applied nowadays. We expect these categories to be most useful for practitioners.

Conceptually, it is certainly interesting to consider the *Effect class*, which is a refinement of Bier et al.'s *Operation class*. Depending on the lens function  $\lambda$ , a lens can suppress irrelevant content, alter existing content, or enrich a visualization with new content. We also discussed the peculiarity of lenses for visualization to show the lens effect



beyond the lens interior. This is captured in the new *Effect extent* category. Another interesting new observation is related to the *Adjustability* of lenses. Properties and parameters of lenses can be non-adjustable or controllable via interaction. Quite intriguing from a research perspective are self-adjusting lenses.

The new categories *Selection  $\sigma$  stage* and *Join  $\bowtie$  stage* establish the connection of lenses to the visualization pipeline. They help to better understand which information is fetched from and is returned to the visualization pipeline in order to create certain lens effects. As discussed in Section 2.3, the flexibility to combine information from the data space (*DS*, *DT*), the model space (*VA*), and the screen space (*V*) considerably broadens the design space of lenses for visualization.

Just like in Bier et al.'s original taxonomy, some of the categories of our extended schema have a finite set of possible characteristics, others are naturally infinite. For example, for the *Effect class*, it is clear that suppress, alter, and enrich are complete. In contrast to that, the *User task* category is clearly an example of being incomplete and open-ended. Currently, Yi et al.'s [YKSJ07] seven interaction intents are sufficient to describe lenses in relation to user tasks. Yet in the future it might be necessary to expand on user tasks, for example, by considering the 32 action patterns by Sedig and Parsons [SP13].

The taxonomy now allows us to organize the existing lens techniques. Table 1 back on page 12 already provides a classification with respect to data and tasks. For a researcher, it could be interesting to create a similar table with respect to *Selection  $\sigma$  stage* and *Join  $\bowtie$  stage*. Yet, such a table is not so easy to compile. While the *Effect class* and the *Effect extent*, for example, are usually easy to figure out (we can see them), selection and join are not. The reason is that the existing literature on lenses often does not provide sufficient in-depth information to pinpoint a technique precisely to particular stage of the visualization pipeline. Yet, this is a difficulty that we share with other taxonomies in visualization, including the famous reference model by Chi [Chi00].

In this sense, our taxonomy may also be useful as an outline according to which new lens techniques can be described in greater detail and more consistently to facilitate reasoning and foster reproducible results. To demonstrate the benefit, we will next dissect a lens technique in depth.

## 6. Detailed Example

So far, we considered conceptual aspects, data and tasks, as well as display and interaction in detail and derived a taxonomy for lenses. Concrete lens techniques were covered only briefly, mainly for the purpose of illustrating our line of thought. In this section, we turn the tables. We will focus on a concrete lens technique, and by means of this technique, summarize and illustrate the relevant aspects as collected in our taxonomy and discussed in the earlier sections.

**Practical Perspective** Let us start with a description of the problem we aim to solve with a lens technique: edge-clutter in node-link diagrams of graph data. Even sophisticated graph drawing algorithms cannot guarantee clutter-free layouts for arbitrary graphs with many nodes and edges. Edge-clutter makes it difficult to identify how many connections a node has, because edges that actually connect to a node cannot be distinguished from edges that just pass by a node by chance. So when the user is interested in determining node connectivity, a sensible approach is to filter out edges that are not relevant at the moment.

A lens technique that supports the task of *filtering* on graph data is the *Local Edge Lens* [TavHS06]. The following discussion draws from a novel implementation of this lens [KRD14] as shown in Figure 16. The lens is designed for a *desktop* scenario where capacitive tracking of *multi-touch* input allows for natural interaction.

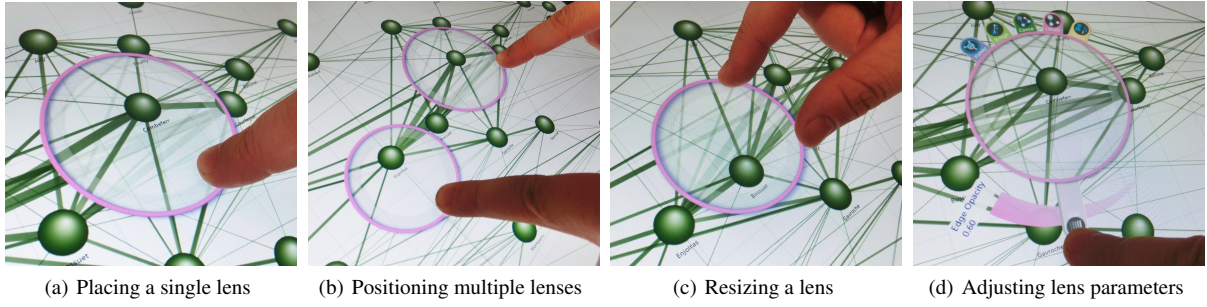
In this setting, the base visualization shows a full-screen node-link diagram of a graph. To facilitate data exploration, nodes and edges are sensitive to direct touch interaction. This sensitivity has to be maintained at all times, even when a lens is active to filter out edges. This is an interesting design challenge. While the lens is supposed to significantly affect the display by filtering edges, it must only minimally interfere with the interactivity of the base visualization. But still, the lens has to support the direct manipulation of its properties.

Using the lens interior as an interaction handle would be too much interference. Yet, using the lens border for lens interaction seem to be a suitable compromise. The border already serves as visual feedback to represent the lens as an interactive tool and to clearly separate lens interior and exterior. Now the border is to be used for interaction as well. As the lens is touch-based, the border has to be sufficiently wide to afford easy picking. Designed this way, the border can be used as an interaction handle for direct manipulation of key lens properties.

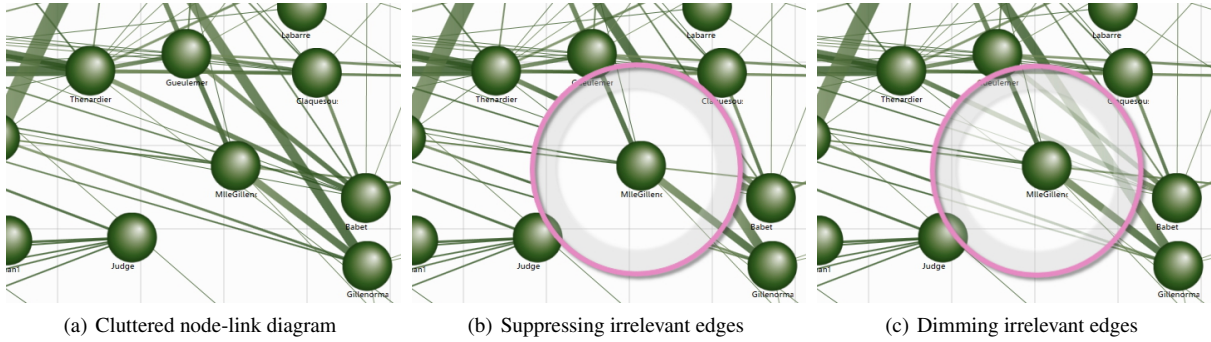
Figures 16(a) and 16(b) illustrate how the position of a single or multiple lenses can be adjusted with a single finger or via bimanual interaction. The size of a lens can be controlled by applying a pinch gesture as shown in Figure 16(c).

In order to support in-situ manipulation of the lens parameters, additional user interface elements are attached to the lens border. Figure 16(d) shows buttons and drag-out sliders along the border. These can be activated and adjusted by easy tap and drag gestures, for example, to control the degree of dimming of filtered-out edges.

**Conceptual Perspective** In order to reduce edge clutter, the lens *selection  $\sigma$*  has to determine which edges are relevant, and which are not relevant and can be filtered out. As we are interested in the connectivity of nodes in a focus region, relevant edges are those that connect to nodes being located inside the lens. Following the classic lens metaphor, inside



**Figure 16:** Touch interaction for lenses. (a) Placing a single lens; (b) Bimanual interaction with multiple lenses; (c) Resizing a lens with a pinch gesture; (d) Direct manipulation user controls at the border of the lens.



**Figure 17:** A graph lens for dealing with edge clutter. (a) Base visualization cluttered with edges; (b) Filtering irrelevant edges by suppressing them inside the lens; (c) Filtering irrelevant edges by dimming them according to a lens parameter.

means contained in a circular *shape* whose *position* and *size* has been set interactively by the user (*lens orientation* is negligible). The inside-criterion can be decided based on the graph layout, which is available in the VA stage of the visualization pipeline. More precisely, at least one node of an edge has to be inside the lens to categorize an edge as relevant. All other edges are irrelevant.

The *lens function*  $\lambda$  now has to filter the irrelevant edges. This goal can be achieved in different ways. We could *suppress* the irrelevant edges inside the lens altogether as in Figure 17(b). This would effectively clear the lens interior of visual stimuli that are not essential to the task at hand. Yet, in some situations it could also be useful have some residues of the irrelevant edges to hint at the fact that there are more than just the relevant ones. As shown in Figure 17(c), this can be achieved by dimming irrelevant edges as they pass the lens, that is, to *alter* the edge representation locally. The dimming can be controlled by a parameter with a value between 0 (invisible) and 1 (visible). Typically this parameter is represented by the alpha channel of the edge color, but could also be related to blurring of the edge stroke.

Depending on which strategy is chosen, there are different ways of implementing the *join*  $\bowtie$ . One option is to realize the

join at the pixel level in screen space ( $V$ ). In a first pass, all edges are rendered to generate the base visualization. In a second pass, only the relevant edges are rendered inside the lens along with the lens shape. The output of the second pass then simply overwrites the output of the first pass.

Alternatively, the join could be realized at the geometry level ( $VA$ ). In this case, the irrelevant edges are split into segments at their intersections with the lens shape. The edge segments that are inside the lens are modified by assigning to them the dimmed rendering style (e.g., the alpha channel). All other edge segments keep their original appearance. The modifications are passed over to the base visualization, which then renders all edge segments (original and modified) in a single pass.

On programmable graphics hardware, it is possible to combine the methods described above via suitable shader programs. When the defining lens properties (i.e., position, size, dimming parameter) are fetched to the shader program, the vertex shader can decide whether an edge is subject to filtering or not. Passing this information down the rendering pipeline enables the fragment shader to alter those pixels of irrelevant edges that are inside the lens.

<i>Practical</i>	
Data type:	Graph
User task:	Filter
Display setting:	Desktop
Interaction modality:	Multi-touch
<i>Conceptual</i>	
Effect class:	Suppress (or alter)
Effect extent:	Lens interior
Adjustability:	Interactive
Selection $\sigma$ stage:	VA
Join $\bowtie$ stage:	V (or VA)

**Table 4:** Categorization of Kister et al.’s [KRD14] multi-touch lens for filtering edge clutter in graph visualizations.

In summary, the discussed lens is a useful technique for focused, lightweight, and temporary clutter reduction in a node-link visualization. Our detailed description of the lens covered all aspects that are relevant in terms of the taxonomy proposed in the previous section. We can now create a profile for the lens, as compiled in Table 4. It remains a task for future work to create such profiles for all lenses in existence. Further topics for future work on lenses for visualization will be discussed in the next section.

## 7. Future Directions

The versatility of our survey indicates that lenses and visualization go very well together. It is only natural that in the future new visualization problems will be answered with new lens techniques. In this section, we want to discuss directions for future research on lenses on a more general level.

**Lenses and Interaction** Virtually all lenses reviewed in this survey support direct manipulation of the lens position on the screen. Adjusting other properties such as size, shape, or parameters controlling the lens function often needs to be carried out via external controls. Maureen Stone, one of the investigators of Magic Lenses, made an important and thought-provoking statement in her review of an initial draft of our survey: “[...] the fundamental problem is how you provide the user a quick and easy way to: Position the lens, work through the lens, and (possibly) parameterize the lens.” We add to her statement the need to flexibly combine lenses to create new lens functions on the fly. Addressing these issues requires more studies on the interaction side of lenses, beyond direct manipulation of their position. Future research has to find new ways of making lenses easy to apply and customize, even when the underlying task is more complicated than simple magnification. As indicated in the previous section, utilizing modern interaction modalities seems to be a promising direction.

**Lenses in Novel Visualization Environments** Existing lens techniques are usually designed for classic desktop visualization settings with mouse and keyboard interaction. We have also seen that first approaches explore the possibilities offered by modern technology, including touch interaction or gaze-based interaction. Moreover, there are lenses that support visualization in environments with multiple or large high-resolution displays. But these are only first steps toward a better utilization of lenses in novel visualization environments. In addition to taking advantage of modern technology, there is also the need to address typical application scenarios in such environments. A particular direction for future work is the investigation of lenses for collaborative multi-user work in future data analytics workspaces for rich, multifaceted exploration and manipulation of information. This involves investigating user-specific private and shared lenses (see [SCG10] and [KRMD15] for first examples) as well as combining individual lenses to a greater whole in a joint effort.

**Lenses for Exploration and Manipulation** The majority of lenses support data exploration in the sense that data is consumed by the user. Yet data manipulation becomes more and more important. Data intensive work often involves data correction, data abstraction, data transformation, or in general terms data manipulation. In such scenarios, the user is the producer of data, a fact that leads to requirements that are different from those for pure exploration tasks. A first lens that addresses the problem of data editing in a visualization setting is the *EditLens* [GSE\*14]. To establish lenses more broadly as tools not only for data exploration, but also for data manipulation, further research is needed. The basic manipulation operations insert, update, and delete must be studied and refined in the context of the different types of data that are relevant for visual data editing. New editing lenses have to take into account the specific characteristics of the data and the requirements of the data manipulation objective. Particularly challenging in this regard are data that are given at multiple scales or at different levels of abstraction.

**Lenses as Universal Tools** Our survey already indicates that lenses are broadly applicable in visualization. In the *big data* era, we expect lenses to become even more important as versatile tools for working with large and complex datasets. However, wide adoption of lens approaches is currently hindered by the lack of a unified toolkit that can be easily integrated with existing visualization approaches. As of today, lens techniques are tightly interwoven with the base visualization, which makes it difficult to transfer existing solutions to related problems. So an important goal for future research is to come up with concepts and methods that facilitate *implement-once-and-reuse-many-times* development of lens techniques. A long-term goal could be to develop lenses that are deployable in terms of lenses as a service to flexibly enrich the capabilities of existing visual interfaces for data-intensive work.

## 8. Conclusion

In this work, we surveyed the literature on interactive lenses in visualization. We introduced a conceptual model of lenses according to which lenses are defined as additional lens pipelines that are attached to a base visualization pipeline. In our model, a visualization lens can take effect at any stage of the transformation from the raw data to their visual representation. The wide applicability of lenses in visualization contexts was illustrated by a systematic review of lens techniques for various types of data and user tasks. Actual designs of interaction for lenses have been described in the light of classic and modern output devices and interaction modalities. Based on the collected knowledge, we extracted a taxonomy of lenses for visualization. A more detailed description of a concrete lens technique for filtering edges in a graph layout completed our survey.

In addition to the review of existing work, we looked into the future and identified promising directions for further research on lenses in visualization. We hope that our survey can stimulate not only a broader utilization of existing lenses, but also motivate researchers to investigate novel concepts and techniques that manifest lenses as rich and flexible tools for visual exploration and manipulation.

## Acknowledgments

We gratefully acknowledge support from the German Research Foundation under grants DA 1319/3-1 and SCHU 887/14-1.

## Appendix: Summary of References

The purpose of this appendix is to share with the reader all the references we used as a basis for our survey. The appendix includes also papers that we did not explicitly cite in the previous sections, but which we think are still valuable sources of information. To go beyond just listing the references, we group them into different categories and provide a brief explanatory caption for each cited paper. We chose an outline according to the following categories: visualization lenses, lens-like techniques, general-purpose magnification and distortion lenses, miscellaneous lenses, models, concepts & taxonomies, and interaction with lenses.

### Visualization Lenses

- [AACP14] RouteLens: Magnification lens that automatically adjusts its position to paths on a map.
- [BCPS02] Fuzzy Lenses: alternative gene representation; Base Pair Lens: separation of genes; Ring Lens: detail in context view applied to genetic network data.
- [BRL09] Enhanced Excentric Labeling: Adaptive lens for labeling data points in scatter plots.
- [CC13] Text visualization assisted by interactive lens queries on spatial models.
- [CMS94] Magic Sphere: Various lens functions for altering 3D visual representations.
- [DMC15] VectorLens: Angular selection and filtering of curves in a 2D visualization.
- [DSA15] 3DArcLens: Deformation and filtering lens for untangling graphs on virtual globes.
- [EBD05,ED06b,ED06a] Sampling Lens: Interactive sampling in scatter plots and parallel coordinates plots.
- [EDF11] Color Lens: Locally adapted color coding for visual representations of data and images.
- [FG98] Lenses for flow data: Magic lens and magic box for higher-resolution streamline visualization.
- [FP99] Excentric Labeling: Labeling points in visual representations of data.
- [GNBP11] FlowLens: Visualizing flow data with additional attributes in a 2D lens.
- [GSE\*14] EditLens: Semi-automatic editing of graphs with customized layouts.
- [HLTE11] SemLens: Analyzing scatter plots with combinable semantic lenses.
- [HTE11] MoleView: Rearranging graphical primitives of multivariate relational data with lenses.
- [JDK10] Network Lens: Representing data attributes in network structures.
- [KCJ\*10] Detail Lenses: Facilitating the understanding of navigation directions via detail maps for points of interest.
- [Kin10] SignalLens: Analyzing details in signal traces with a focus+context approach.
- [KRD14] Multi-touch interaction for working with multiple lenses on graphs.
- [KSW06] ClearView: An illustrative lens for focus+context exploration of volume data sets.
- [KTW\*13] TrajectoryLenses: Lenses for filtering and exploring trajectory data.
- [LAM10] Lenses for enhancing the exploration edge-bundled graphs.
- [LHJ01] Lenses for magnifying focus regions in volume data.
- [LWG11] Facet Lens: Exploration of document data with multiple facets.
- [MCH\*09] Bring & Go: Rearranging graph vertices for quick navigation.
- [MTHG03] Magic Volumes: Focus+context visualization of 3D flow data.
- [PBKE11] EdgeAnalyser: Support edge grouping and insight into properties of groups of edges.
- [PPCP12] JellyLens: Lens that adapts its shape to features in visual representations.
- [PPCP13] Gimlenses: Drilling holes into complex 3D models to uncover details.
- [RC94] Table Lens: Focus+context for tabular visualizations.
- [RHS05] Lenses for focus+context in geo-virtual environments.
- [RLE05] Temporal Magic Lens: Combining spatial and temporal aspects for visual analysis of video data.



- [RM93] Document Lens: Classic lens for document visualization.
- [SB92, SB94] Fisheye Views: Classic focus+context with application to visualization.
- [SFR01] Time Lens: Comparing different points in time for understanding spatio-temporal data.
- [SHER99] Interactive 3D lens: Enhancing 3D glyph-based visual representations.
- [SNDC10] PushLens: Routing edges around the lens to reduce ambiguities in graph layouts.
- [STSD10] Tangible views: Interactive visualization lenses made tangible.
- [TAvHS06, TAS09] Graph lenses: Supporting graph exploration through lens techniques.
- [TFS08] Smart lenses: Integrating lenses into the data state reference model of visualization.
- [TGBD08] 3D Generalization Lenses: Focus+context visualization of virtual city models.
- [TSA12] Time Lens: Enhancing the visualization of spatial movements with temporal aspects.
- [UvK13] Magic lenses for hypergraphs: Querying graph structures with hyperedges using interactive lenses.
- [VCWP96] 3D Magic lenses: Extending magic lenses to the third dimension.
- [vHvW04] Graph abstraction lens: Combining semantic and geometric distortion for analyzing hierarchical graphs.
- [WCG03] EdgeLens: Bending graph edges around a lens to reduce ambiguities.
- [WZMK05] The Magic Volume Lens: Magnification of features of interest in volume visualization.
- [ZCB11] MagicAnalytics Lens: On-the-fly comparison of time series regarding analytical methods.
- [ZCPB11] ChronoLenses: Transform time series on the fly.

### Lens-Like and Illustrative Techniques

- [CSC06] Peel-away visualization to enable insight into volume data.
- [BDW\*08] Probes for geospatial analysis: Access to details by placing probes on maps.
- [BG06] Illustrative visualization using exploded views of volumetric data.
- [BGM\*10] Illustrative focus+context: Cutaway views and ghosting for volume visualization.
- [BNP\*13] Transmogrifiers enable on-the-fly graphic transformations using flexible shapes.
- [Elm05] BalloonProbe: Interactive reduction of occlusion problems in 3D visualization scenes.
- [FS92] Cutaways and ghosting: Addressing visibility problems in 3D illustrations.
- [HS04] Timeboxes: Movable filter tools for querying time series.
- [KFA\*04] Frisbee: Remote control interface for large displays.
- [LRA\*07] Cutaway illustrations of complex 3D models.

- [MRC91] Classic detail and context approach using a 3D perspective projection.
- [MTB03] Combining cut into and open up, spread apart, or peel away for looking into the interior of volumes.
- [VKG04] Importance-Driven Volume Rendering: Cutaway and ghosting for focus+context.
- [WEE03] Interactive clipping of visualizations of volumetric data.

### General-Purpose Magnification and Distortion Lenses

- [ACP10] High precision magnification lenses: Improved picking through separation of visual and motor space.
- [CLB\*16] SchemeLens: Topology-aware distortion for navigating large system diagrams.
- [BCN11] Undistort lens: Locally undistort representations that have been modified by global distortion techniques.
- [BEK\*03] Screen magnifiers: A review of techniques for magnifying screen content.
- [BHR14] PhysicLenses: Multifocus magnification using a physical collision model and touch interaction.
- [CLP04] Higher magnification: New functions that improve magnification while maintaining context.
- [FS05] DTLens: Multi-user, multi-lens magnification on tabletops.
- [GGSS06] SpringLens: Flexible distributed nonlinear magnification using a mass-spring model.
- [Gut02] Speed-coupled flattening: Improving focus targeting in fisheye lenses.
- [MW14] Bubble Lens: Increasing speed and accuracy of small and dense target selection.
- [PA08, PBA10] Sigma lenses: Using space, translucence, and time for transitions between focus and context.
- [RCBBL07] Pointing Lenses: Easier selection of small targets.
- [SG07] Using fisheye lenses to support multi-focus interaction.
- [ZZG\*12] Conformal magnifier: Magnification with conformal mapping and local shape preservation.

### Miscellaneous Lenses

- [HRS97] Debugging Lenses: Supporting debugging with lens techniques.
- [KMS09] AR focus+context visualization: Lenses restrict overlaid context presentation.
- [NIC07] NPR Lenses: Lenses with non-photorealistic rendering effects.
- [SCG10] IdLenses: Personalization of graphical content through lenses.

### Models, Concepts & Taxonomies

- [BSF\*94] A taxonomy of the See-Through Interface, including Toolglass and Magic Lenses.
- [BSP\*93] The classic paper on Toolglass and Magic Lenses.

- [BSP97] Various Magic Lens filters for illustration systems.
- [CCF97] Conceptual considerations when extending 2D distortion to 3D distortion.
- [CKB08] Survey on overview+detail, zooming, and focus+context approaches.
- [CM01] A unified model for presentation spaces allowing for different presentation methods in a single interface.
- [CMS99] Model of the visualization process as a pipeline of data transformation stages.
- [EBZ\*12] Design considerations for workspaces with large high-resolution displays.
- [Fox98] A generic model for composing multiple lens functions via delegation, rather than inheritance.
- [FS95] Composing movable filters for enhanced dynamic querying.
- [Fur86] The theory behind degree of interest for focus+context approaches.
- [Hau06] A generalized view on focus+context in visualization.
- [Kea98] Generalization of the focus within context problem.
- [KR96] Functions for non-linear magnification transformations.
- [LA94] A survey on distortion-oriented presentation techniques.
- [Neu03] A taxonomy and review of discrete lens techniques.
- [RH04] An implementation model for 3D lenses.
- [Rob07] Coordinated multiple views for exploratory visualization.
- [SFB94] The concept of the movable filter as an interface tool.
- [Shn83] The classic definition of interaction through direct manipulation.
- [Shn96] Taxonomy of visualization based on data types and tasks.
- [SSS02] Taxonomy of the through-the-lens metaphor.
- [Tom11] A model for event-based concepts and visualization.
- [WGK10] A textbook explaining foundations, techniques, and applications of visualization.
- [YCB05] Models for nonlinear perspective distortion and lenses for 3D scenes.
- [YKSJ07] Categorization of interaction techniques according to user intents.
- [BKFD14] Analysis of transparent tangibles on tabletops and their advantages for precision.
- [EWCP13] TangibleRings as intuitive tangible interfaces for nestable lenses.
- [IIH\*13] A research agenda for visualization on interactive surfaces.
- [IU97] Tangible interaction to bridge the gap between cyberspace and physical world.
- [Jac90] Eye movement-based interaction techniques.
- [KAP11] FingerGlass: Interactive, bimanual touch-based lens interaction via ring objects.
- [KE12] Embodied lenses: physical interaction with sheets of foil with applications to multi-dimensional data visualization.
- [KPW07] Eyepoint: Gaze and keyboard interaction for magnification lenses.
- [KRMD15] BodyLenses: Lenses that are controlled by body movement and posture in front of wall displays.
- [Lan00] Magnification lenses in gaze-based interaction.
- [LBC04] Using tangibles as handles for virtual lenses in AR.
- [LG14] Combining a cloth metaphor and touch interaction for multi-user content exploration.
- [LGB07] Tangible lens with flexible bendable shape in AR.
- [LSST11] Head-tracking for lens interactions in front of large high-resolution displays.
- [RK14] Multi-touch interaction for data visualization, physics-based manipulations.
- [SBD12] Head-coupled perspective with tangible views.
- [SD13] Zones for gaze-based interaction with magnification lenses.
- [SS14] Touch interaction for magnification lenses in a scatterplot visualization.
- [SSD09, SBWD14] PaperLens and tangible displays: passive and active tangible views tracked above the tabletop.
- [SSND11] Multimodal gaze-supported interaction.
- [SSRG11] Feature tracking with lens-like handheld devices for large display walls.
- [SSTR93] Interactive view transformation using a rubber sheet metaphor.
- [TPH14] Display deformation used to create magnification lenses.
- [VTS\*09] Design considerations for tabletop interaction techniques.
- [WKS07] Optical Magic Lenses: Using real lenses to control virtual lenses.

## Lens interaction

- [ADS05] Improving gaze-based interaction with the help of a fisheye lens.
- [BLC12] Modern multi-touch techniques for interacting with stacked graphs of multivariate time-series.
- [BWB06] Improved selection of small targets on multi-touch screens with dual finger interaction techniques.

## References

- [AAP14] ALVINA J., APPERT C., CHAPUIS O., PIETRIGA E.: Routelens: Easy route following for map applications. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2014), ACM, pp. 125–128. doi:10.1145/2598153.2598200. 7, 12, 13, 20
- [ACP10] APPERT C., CHAPUIS O., PIETRIGA E.: High-Precision Magnification lenses. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2010), ACM,

- pp. 273–282. doi:10.1145/1753326.1753366. 12, 13, 21
- [ADS05] ASHMORE M., DUCHOWSKI A. T., SHOEMAKER G.: Efficient Eye Pointing with a Fisheye Lens. In *Proc. of Graphics Interface Conference (GI)* (2005), Canadian Human-Computer Communications Society, pp. 203–210. URL: <http://dl.acm.org/citation.cfm?id=1089508.1089542>. 15, 22
- [BCN11] BROSZ J., CARPENDALE M. S. T., NACENTA M. A.: The Undistort Lens. *Computer Graphics Forum* 30, 3 (2011), 881–890. doi:10.1111/j.1467-8659.2011.01937.x. 21
- [BCPS02] BAKER C. A. H., CARPENDALE M. S. T., PRUSINKIEWICZ P., SURETTE M. G.: GeneVis: Visualization Tools for Genetic Regulatory Network Dynamics. In *Proc. of IEEE Visualization Conference (Vis)* (2002), IEEE Computer Society, pp. 243–250. doi:10.1109/VISUAL.2002.1183781. 12, 13, 20
- [BDW\*08] BUTKIEWICZ T., DOU W., WARTELL Z., RIBARSKY W., CHANG R.: Multi-Focused Geospatial Analysis Using Probes. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1165–1172. doi:10.1109/TVCG.2008.149. 21
- [BEK\*03] BLENKHORN P., EVANS G., KING A., HASTUTI KURNIAWAN S., SUTCLIFFE A.: Screen Magnifiers: Evolution and Evaluation. *Computer Graphics and Applications* 19, 5 (2003), 54–61. doi:10.1109/MCG.2003.1231178. 21
- [Ber81] BERTIN J.: *Graphics and Graphic Information-Processing*. de Gruyter, 1981. 1
- [BG06] BRUCKNER S., GRÖLLER M. E.: Exploded Views for Volume Data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 1077–1084. doi:10.1109/TVCG.2006.140. 3, 21
- [BGM\*10] BRUCKNER S., GRÖLLER M. E., MUELLER K., PREIM B., SILVER D.: Illustrative Focus+Context Approaches in Interactive Volume Visualization. In *Scientific Visualization: Advanced Concepts*, Hagen H., (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010, pp. 136–162. doi:10.4230/DFU.SciViz.2010.136. 21
- [BHR14] BUTSCHER S., HORNBAEK K., REITERER H.: Space-Fold and PhysicLenses: Simultaneous Multifocus Navigation on Touch Surfaces. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2014), ACM, pp. 209–216. doi:10.1145/2598153.2598177. 12, 14, 21
- [BKFD14] BÜSCHEL W., KISTER U., FRISCH M., DACHSELT R.: T4 – Transparent and Translucent Tangibles on Tabletops. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2014), ACM, pp. 81–88. doi:10.1145/2598153.2598179. 15, 22
- [BLC12] BAUR D., LEE B., CARPENDALE S.: TouchWave: Kinetic Multi-touch Manipulation for Hierarchical Stacked Graphs. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2012), ACM, pp. 255–264. doi:10.1145/2396636.2396675. 13, 22
- [BLT\*12] BARIČEVIĆ D., LEE C., TURK M., HÖLLERER T., BOWMAN D. A.: A Hand-Held AR Magic Lens with User-Perspective Rendering. In *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2012), IEEE Computer Society, pp. 197–206. doi:10.1109/ISMAR.2012.6402557. 15
- [BNP\*13] BROSZ J., NACENTA M. A., PUSCH R., CARPENDALE S., HURTER C.: Transmogrification: Causal Manipulation of Visualizations. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2013), ACM, pp. 97–106. doi:10.1145/2501988.2502046. 21
- [BRL09] BERTINI E., RIGAMONTI M., LALANNE D.: Extended Excentric Labeling. *Computer Graphics Forum* 28, 3 (2009), 927–934. doi:10.1111/j.1467-8659.2009.01456.x. 4, 5, 7, 12, 13, 20
- [BSF\*94] BIER E. A., STONE M. C., FISHKIN K. P., BUXTON W., BAUDEL T.: A Taxonomy of See-Through Tools. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1994), ACM, pp. 358–364. doi:10.1145/191666.191786. 2, 16, 21
- [BSP\*93] BIER E. A., STONE M. C., PIER K., BUXTON W., DEROSE T. D.: Toolglass and Magic Lenses: the See-Through Interface. In *Proc. of Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (1993), ACM, pp. 73–80. doi:10.1145/166117.166126. 2, 3, 21
- [BSP97] BIER E., STONE M., PIER K.: Enhanced Illustration Using Magic Lens Filters. *Computer Graphics and Applications* 17, 6 (1997), 62–70. doi:10.1109/38.626971. 2, 22
- [BWB06] BENKO H., WILSON A. D., BAUDISCH P.: Precise selection techniques for multi-touch screens. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2006), ACM, pp. 1263–1272. doi:10.1145/1124772.1124963. 14, 22
- [CC13] CHANG M.-W., COLLINS C.: Exploring Entities in Text with Descriptive Non-Photorealistic Rendering. In *Proc. of IEEE Pacific Visualization Symposium (PacificVis)* (2013), IEEE Computer Society, pp. 9–16. doi:10.1109/PacificVis.2013.6596122. 9, 12, 20
- [CCF97] CARPENDALE M., COWPERTHWAIT D., FRACCHIA F.: Extending Distortion Viewing from 2D to 3D. *Computer Graphics and Applications* 17, 4 (1997), 42–51. doi:10.1109/38.595268. 7, 22
- [Chi00] CHI E. H.: A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *Proc. of IEEE Symposium Information Visualization (InfoVis)* (2000), IEEE Computer Society, pp. 69–75. doi:10.1109/INFVIS.2000.885092. 17
- [CKB08] COCKBURN A., KARLSON A., BEDERSON B. B.: A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Computing Surveys* 41, 1 (2008), 2:1–2:31. doi:10.1145/1456650.1456652. 1, 22
- [CLB\*16] COHÉ A., LIUTKUS B., BAILLY G., EAGAN J., LECOLINET E.: SchemeLens: A Content-Aware Vector-Based Fisheye Technique for Navigating Large Systems Diagrams. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 330–338. doi:10.1109/TVCG.2015.2467035. 21
- [CLP04] CARPENDALE S., LIGH J., PATTISON E.: Achieving Higher Magnification in Context. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2004), ACM, pp. 71–80. doi:10.1145/1029632.1029645. 8, 21
- [CM01] CARPENDALE M. S. T., MONTAGNESE C.: A Framework for Unifying Presentation Space. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2001), ACM, pp. 61–70. doi:10.1145/502348.502358. 8, 22
- [CMS94] CIGNONI P., MONTANI C., SCOPIGNO R.: MagicSphere: an Insight Tool for 3D Data Visualization. *Computer Graphics Forum* 13, 3 (1994), 317–328. doi:10.1111/1467-8659.1330317. 12, 20

- [CMS99] CARD S., MACKINLAY J., SHNEIDERMAN B.: *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999. 3, 22
- [CSC06] CORREA C. D., SILVER D., CHEN M.: Discontinuous Displacement Mapping for Volume Graphics. In *Proc. of Eurographics/IEEE VGTC Workshop on Volume Graphics (VG)* (2006), Eurographics Association, pp. 9–16. doi:10.2312/VG/VG06/009–016. 3, 21
- [DMC15] DUMAS M., MCGUFFIN M., CHASSÉ P.: VectorLens: Angular Selection of Curves within 2D Dense Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 21, 3 (2015), 402–412. doi:10.1109/TVCG.2014.2362543. 12, 20
- [DSA15] DEBIASI A., SIMÕES B., AMICIS R. D.: 3DArcLens: Interactive Network Analysis on Geographic Surfaces. In *Proc. of International Conference on Information Visualization Theory and Applications* (2015), SCITEPRESS, pp. 291–299. doi:10.5220/0005255202910299. 12, 20
- [EBD05] ELLIS G., BERTINI E., DIX A.: The Sampling Lens: Making Sense of Saturated Visualisations. In *CHI Extended Abstracts on Human Factors in Computing Systems* (2005), ACM, pp. 1351–1354. doi:10.1145/1056808.1056914. 9, 12, 13, 20
- [EBZ\*12] ENDERT A., BRADEL L., ZEITZ J., ANDREWS C., NORTH C.: Designing Large High-resolution Display Workspaces. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2012), ACM, pp. 58–65. doi:10.1145/2254556.2254570. 11, 22
- [ED06a] ELLIS G., DIX A. J.: Enabling Automatic Clutter Reduction in Parallel Coordinate Plots. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 717–724. doi:10.1109/TVCG.2006.138. 9, 12, 13, 20
- [ED06b] ELLIS G., DIX A. J.: The Plot, the Clutter, the Sampling and its Lens: Occlusion Measures for Automatic Clutter Reduction. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2006), ACM, pp. 266–269. doi:10.1145/1133265.1133318. 5, 7, 9, 12, 20
- [EDF11] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Color Lens: Adaptive Color Scale Optimization for Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics* 17, 6 (2011), 795–807. doi:10.1109/TVCG.2010.94. 10, 11, 12, 20
- [Elm05] ELMQVIST N.: BalloonProbe: Reducing Occlusion in 3D Using Interactive Space Distortion. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)* (2005), ACM, pp. 134–137. doi:10.1145/1101616.1101643. 21
- [EWCP13] EBERT A., WEBER C., CERNEA D., PETSCH S.: TangibleRings: Nestable Circular Tangibles. In *CHI Extended Abstracts on Human Factors in Computing Systems* (2013), ACM, pp. 1617–1622. doi:10.1145/2468356.2468645. 13, 14, 22
- [FG98] FUHRMANN A., GRÖLLER E.: Real-time Techniques for 3D Flow Visualization. In *Proc. of IEEE Visualization Conference (Vis)* (1998), IEEE Computer Society, pp. 305–312. doi:10.1109/VISUAL.1998.745317. 7, 8, 9, 12, 13, 20
- [Fox98] FOX D.: Composing Magic Lenses. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1998), ACM, pp. 519–525. doi:10.1145/274644.274714. 8, 22
- [FP99] FEKETE J.-D., PLAISANT C.: Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1999), ACM, pp. 512–519. doi:10.1145/302979.303148. 12, 20
- [FS92] FEINER S. K., SELIGMANN D. D.: Cutaways and Ghosting: Satisfying Visibility Constraints in Dynamic 3D Illustrations. *The Visual Computer* 8, 5–6 (1992), 292–302. doi:10.1007/BF01897116. 3, 21
- [FS95] FISHKIN K. P., STONE M. C.: Enhanced Dynamic Queries via Movable Filters. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1995), ACM, pp. 415–420. doi:10.1145/223904.223960. 2, 22
- [FS05] FORLINES C., SHEN C.: DTLenS: Multi-user Tabletop Spatial Data Exploration. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2005), ACM, pp. 119–122. doi:10.1145/1095034.1095055. 8, 21
- [Fur86] FURNAS G. W.: Generalized Fisheye Views. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1986), ACM, pp. 16–23. doi:10.1145/22627.22342. 2, 22
- [GGSS06] GERMER T., GÖTZELMANN T., SPINDLER M., STROTHOTTE T.: SpringLens: Distributed Nonlinear Magnifications. In *Eurographics - Short Papers* (2006), Eurographics Association, pp. 123–126. 21
- [GNBP11] GASTEIGER R., NEUGEBAUER M., BEUING O., PREIM B.: The FLOWLENS: A Focus-and-Context Visualization Approach for Exploration of Blood Flow in Cerebral Aneurysms. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2183–2192. doi:10.1109/TVCG.2011.243. 7, 8, 12, 20
- [GSE\*14] GLADISCH S., SCHUMANN H., ERNST M., FÜLLEN G., TOMINSKI C.: Semi-Automatic Editing of Graphs with Customized Layouts. *Computer Graphics Forum* 33, 3 (2014), 381–390. doi:10.1111/cgf.12394. 19, 20
- [Gut02] GUTWIN C.: Improving Focus Targeting in Interactive Fisheye Views. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2002), ACM, pp. 267–274. doi:10.1145/503376.503424. 21
- [Hau06] HAUSER H.: Generalizing Focus+Context Visualization. In *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Bonneau G.-P., Ertl T., Nielson G., (Eds.). Springer, 2006, pp. 305–327. doi:10.1007/3-540-30790-7\_18. 1, 22
- [HLTE11] HEIM P., LOHMANN S., TSENDRAGCHAA D., ERTL T.: SemLens: Visual Analysis of Semantic Data with Scatter Plots and Semantic Lenses. In *Proc. of International Conference on Semantic Systems* (2011), ACM, pp. 175–178. doi:10.1145/2063518.2063543. 12, 20
- [HRS97] HUDSON S. E., RODENSTEIN R., SMITH I.: Debugging Lenses: A New Class of Transparent Tools for User Interface Debugging. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (1997), ACM, pp. 179–187. doi:10.1145/263407.263542. 21
- [HS04] HOCHHEISER H., SHNEIDERMAN B.: Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization* 3, 1 (2004), 1–18. doi:10.1145/993176.993177. 21
- [HTE11] HURTER C., TELEA A., ERSOY O.: MoleView: An Attribute and Structure-Based Semantic Lens for Large Element-Based Plots. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2600–2609. doi:10.1109/TVCG.2011.223. 12, 13, 20
- [IHH\*13] ISENBERG P., ISENBERG T., HESSELMANN T., LEE B., VON ZADOW U., TANG A.: Data Visualization on Interactive Surfaces: A Research Agenda. *Computer Graphics and Applications* 33, 2 (2013), 16–24. doi:10.1109/MCG.2013.24. 11, 22



- [IU97] ISHII H., ULLMER B.: Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1997), ACM, pp. 234–241. doi:10.1145/258549.258715. 14, 22
- [Jac90] JACOB R. J. K.: What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1990), ACM, pp. 11–18. doi:10.1145/97243.97246. 15, 22
- [JDK10] JUSUFI I., DINGJIE Y., KERREN A.: The Network Lens: Interactive Exploration of Multivariate Networks Using Visual Filtering. In *Proc. of International Conference Information Visualisation (IV)* (2010), IEEE Computer Society, pp. 35–42. doi:10.1109/IV.2010.15. 12, 13, 20
- [KAP11] KÄSER D. P., AGRAWALA M., PAULY M.: Finger-Glass: Efficient Multiscale Interaction on Multitouch Screens. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2011), ACM, pp. 1601–1610. doi:10.1145/1978942.1979175. 13, 14, 22
- [KCJ\*10] KARNICK P., CLINE D., JESCHKE S., RAZDAN A., WONKA P.: Route Visualization Using Detail Lenses. *IEEE Transactions on Visualization and Computer Graphics* 16, 2 (2010), 235–247. doi:10.1109/TVCG.2009.65. 8, 9, 12, 20
- [KE12] KIM K., ELMQVIST N.: Embodied Lenses for Collaborative Visual Queries on Tabletop Displays. *Information Visualization* 11, 4 (2012), 319–338. doi:10.1177/1473871612441874. 14, 22
- [Kea98] KEAHEY A.: The Generalized Detail-In-Context Problem. In *Proc. of IEEE Symposium Information Visualization (InfoVis)* (1998), IEEE Computer Society, pp. 44–51. doi:10.1109/INFVIS.1998.729558. 8, 22
- [KFA\*04] KHAN A., FITZMAURICE G. W., ALMEIDA D., BURTNYK N., KURTENBACH G.: A Remote Control Interface for Large Displays. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2004), ACM, pp. 127–136. doi:10.1145/1029632.1029655. 21
- [Kin10] KINCAID R.: SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 900–907. doi:10.1109/TVCG.2010.193. 6, 12, 20
- [KMS09] KALKOFEN D., MÉNDEZ E., SCHMALSTIEG D.: Comprehensible Visualization for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 15, 2 (2009), 193–204. doi:10.1109/TVCG.2008.96. 6, 21
- [KPW07] KUMAR M., PAEPCKE A., WINOGRAD T.: Eye-Point: Practical Pointing and Selection Using Gaze and Keyboard. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2007), ACM, pp. 421–430. doi:10.1145/1240624.1240692. 15, 22
- [KR96] KEAHEY A., ROBERTSON E. L.: Techniques for Non-linear Magnification Transformations. In *Proc. of IEEE Symposium Information Visualization (InfoVis)* (1996), IEEE Computer Society, pp. 38–45. doi:10.1109/INFVIS.1996.559214. 22
- [KRD14] KISTER U., REIPSCHLÄGER P., DACHSELT R.: Multi-Touch Manipulation of Magic Lenses for Information Visualization. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2014), ACM, pp. 431–434. doi:10.1145/2669485.2669528. 12, 17, 19, 20
- [KRMD15] KISTER U., REIPSCHLÄGER P., MATULIC F., DACHSELT R.: BodyLenses: Embodied Magic Lenses and Personal Territories for Wall Displays. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2015), ACM, pp. 117–126. doi:10.1145/2817721.2817726. 6, 19, 22
- [KSW06] KRÜGER J. H., SCHNEIDER J., WESTERMANN R.: ClearView: An Interactive Context Preserving Hotspot Visualization Technique. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 941–948. doi:10.1109/TVCG.2006.124. 12, 20
- [KTW\*13] KRÜGER R., THOM D., WÖRNER M., BOSCH H., ERTL T.: TrajectoryLenses - A Set-based Filtering and Exploration Technique for Long-term Trajectory Data. *Computer Graphics Forum* 32, 3 (2013), 451–460. doi:10.1111/cgf.12132. 10, 11, 12, 20
- [LA94] LEUNG Y. K., APPERLEY M. D.: A Review and Taxonomy of Distortion-oriented Presentation Techniques. *ACM Transactions on Computer-Human Interaction* 1, 2 (1994), 126–160. doi:10.1145/180171.180173. 1, 2, 3, 22
- [LAM10] LAMBERT A., AUBER D., MELANÇON G.: Living Flows: Enhanced Exploration of Edge-Bundled Graphs Based on GPU-Intensive Edge Rendering. In *Proc. of International Conference Information Visualisation (IV)* (2010), ACM, pp. 523–530. doi:10.1109/IV.2010.78. 20
- [Lan00] LANKFORD C.: Effective Eye-gaze Input into Windows. In *Proc. of Symposium on Eye Tracking Research & Applications* (2000), ACM, pp. 23–27. doi:10.1145/355017.355021. 15, 22
- [LBC04] LOOSER J., BILLINGHURST M., COCKBURN A.: Through the Looking Glass: The Use of Lenses As an Interface Tool for Augmented Reality Interfaces. In *Proc. of International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)* (2004), ACM, pp. 204–211. doi:10.1145/988834.988870. 15, 22
- [LG14] LANDER C., GEHRING S.: ClothLens Demo: Simultaneous Multi-User Interaction with Shared Content on a Tabletop. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2014), ACM, pp. 415–418. doi:10.1145/2669485.2669524. 22
- [LGB07] LOOSER J., GRASSET R., BILLINGHURST M.: A 3D Flexible and Tangible Magic Lens in Augmented Reality. In *Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2007), IEEE Computer Society, pp. 51–54. doi:10.1109/ISMAR.2007.4538825. 7, 22
- [LHJ01] LAMAR E., HAMANN B., JOY K.: A magnification lens for interactive volume visualization. In *Proc. of Pacific Conference on Computer Graphics and Applications (PCCGA)* (2001), IEEE Computer Society, pp. 223–232. doi:10.1109/PCCGA.2001.962877. 7, 12, 20
- [LRA\*07] LI W., RITTER L., AGRAWALA M., CURLESS B., SALESIN D.: Interactive Cutaway Illustrations of Complex 3D Models. *ACM Transactions on Graphics* 26, 3 (2007), 31–1–31–11. doi:10.1145/1276377.1276416. 21
- [LSST11] LEHMANN A., SCHUMANN H., STAADT O., TOMINSKI C.: Physical Navigation to Support Graph Exploration on a Large High-Resolution Display. In *Advances in Visual Computing*, Bebis G., Boyle R., Parvin B., Koracin D., Wang S., Kyungham K., Benes B., Moreland K., Borst C., DiVerdi S., Yi-Jen C., Ming J., (Eds.), vol. 6938 of *Lecture Notes in Computer Science*. Springer, 2011, pp. 496–507. doi:10.1007/978-3-642-24028-7\_46. 15, 22
- [LWG11] LAMBECK C., WOJZIAK J., GROH R.: Facet Lens: Local Exploration and Discovery in Globally Faceted Data Sets. In *Proc. of Conference on Creativity and Innovation in Design (DESIRE)* (2011), ACM, pp. 85–88. doi:10.1145/2079216.2079228. 12, 20

- [MCH\*09] MOSCOVICH T., CHEVALIER F., HENRY N., PIETRIGA E., FEKETE J.-D.: Topology-aware Navigation in Large Networks. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2009), ACM, pp. 2319–2328. doi:10.1145/1518701.1519056. 10, 11, 12, 20
- [MRC91] MACKINLAY J. D., ROBERTSON G. G., CARD S. K.: The Perspective Wall: Detail and Context Smoothly Integrated. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1991), ACM, pp. 173–176. doi:10.1145/108844.108870. 2, 21
- [MTB03] MCGUFFIN M. J., TANCAU L., BALAKRISHNAN R.: Using Deformations for Browsing Volumetric Data. In *Proc. of IEEE Visualization Conference (Vis)* (2003), IEEE Computer Society, pp. 401–408. doi:10.1109/VISUAL.2003.1250400. 21
- [MTHG03] MATTAUSCH O., THEUSSL T., HAUSER H., GRÖLLER E.: Strategies for Interactive Exploration of 3D Flow Using Evenly-spaced Illuminated Streamlines. In *Proc. of Spring Conference on Computer Graphics (SCCG)* (2003), ACM, pp. 213–222. doi:10.1145/984952.984987. 12, 20
- [MW14] MOTT M. E., WOBROCK J. O.: Beating the Bubble: Using Kinematic Triggering in the Bubble Lens for Acquiring Small, Dense Targets. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2014), ACM, pp. 733–742. doi:10.1145/2556288.2557410. 10, 12, 21
- [Neu03] NEUMANN P.: *Taxonomy for Discrete Lenses*. Technical report no. 2003-734-37, Department of Computer Science, University of Calgary, 2003. 22
- [NIC07] NEUMANN P., ISENBERG T., CARPENDALE M. S. T.: NPR Lenses: Interactive Tools for Non-Photorealistic Line Drawings. In *Proc. of Smart Graphics (SG)* (2007), Springer, pp. 10–22. doi:10.1007/978-3-540-73214-3\_2. 21
- [PA08] PIETRIGA E., APPERT C.: Sigma Lenses: Focus-Context Transitions Combining Space, Time and Translucence. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2008), ACM, pp. 1343–1352. doi:10.1145/1357054.1357264. 21
- [PBA10] PIETRIGA E., BAU O., APPERT C.: Representation-Independent In-Place Magnification with Sigma Lenses. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (2010), 455–467. doi:10.1109/TVCG.2009.98. 21
- [PBKE11] PANAGIOTIDIS A., BOSCH H., KOCH S., ERTL T.: EdgeAnalyzer: Exploratory Analysis through Advanced Edge Interaction. In *Proc. of Hawaii International Conference on System Sciences (HICSS)* (2011), IEEE Computer Society. doi:10.1109/HICSS.2011.178. 12, 20
- [PPCP12] PINDAT C., PIETRIGA E., CHAPUIS O., PUECH C.: JellyLens: Content-aware Adaptive Lenses. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (2012), ACM, pp. 261–270. doi:10.1145/2380116.2380150. 6, 8, 12, 13, 20
- [PPCP13] PINDAT C., PIETRIGA E., CHAPUIS O., PUECH C.: Drilling into Complex 3D Models with Gimlenses. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)* (2013), ACM, pp. 223–230. doi:10.1145/2503713.2503714. 12, 13, 20
- [RC94] RAO R., CARD S. K.: The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1994), ACM, pp. 318–322. doi:10.1145/191666.191776. 7, 12, 20
- [RCBBL07] RAMOS G., COCKBURN A., BALAKRISHNAN R., BEAUDOUIN-LAFON M.: Pointing Lenses: Facilitating Stylus Input Through Visual-and Motor-space Magnification. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2007), ACM, pp. 757–766. doi:10.1145/1240624.1240741. 21
- [RH04] ROPINSKI T., HINRICHS K.: Real-Time Rendering of 3D Magic Lenses Having Arbitrary Convex Shapes. In *Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)* (2004), UNION Agency - Science Press, pp. 379–386. 7, 22
- [RHS05] ROPINSKI T., HINRICHS K. H., STEINICKE F.: A Solution for the Focus and Context Problem in Geo-Virtual Environments. In *Proc. of ISPRS Workshop on Dynamic and Multi-dimensional GIS (DMGIS)* (2005), pp. 144–149. 12, 20
- [RK14] RZESZOTARSKI J. M., KITTUR A.: Kinetica: Naturalistic Multi-touch Data Visualization. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2014), ACM, (to appear). 14, 22
- [RLE05] RYALL K., LI Q., ESENTHER A.: Temporal Magic Lens: Combined Spatial and Temporal Query and Presentation. In *Proc. of IFIP TC13 Conference on Human-Computer Interaction (INTERACT)* (2005), Springer, pp. 809–822. doi:10.1007/11555261\_64. 12, 20
- [RM93] ROBERTSON G. G., MACKINLAY J. D.: The Document Lens. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (1993), ACM, pp. 101–108. doi:10.1145/168642.168652. 12, 13, 21
- [Rob07] ROBERTS J.: State of the Art: Coordinated Multiple Views in Exploratory Visualization. In *Proc. of International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV)* (2007), IEEE Computer Society, pp. 61–71. doi:10.1109/CMV.2007.20. 1, 22
- [Sam06] SAMET H.: *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006. 4
- [SB92] SARKAR M., BROWN M. H.: Graphical Fisheye Views of Graphs. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1992), ACM, pp. 83–91. doi:10.1145/142750.142763. 12, 21
- [SB94] SARKAR M., BROWN M. H.: Graphical Fisheye Views. *Communications of the ACM* 37, 12 (1994), 73–83. doi:10.1145/198366.198384. 12, 21
- [SBD12] SPINDLER M., BÜSCHEL W., DACHSELT R.: Use Your Head: Tangible Windows for 3D Information Spaces in a Tabletop Environment. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2012), ACM, pp. 245–254. doi:10.1145/2396636.2396674. 15, 22
- [SBWD14] SPINDLER M., BÜSCHEL W., WINKLER C., DACHSELT R.: Tangible Displays for the Masses: Spatial Interaction with Handheld Displays by Using Consumer Depth Cameras. *Personal and Ubiquitous Computing* 18, 5 (2014), 1213–1225. doi:10.1007/s00779-013-0730-7. 15, 22
- [SCG10] SCHMIDT D., CHONG M. K., GELLERSEN H.: IdLenses: Dynamic Personal Areas on Shared Surfaces. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2010), ACM, pp. 131–134. doi:10.1145/1936652.1936678. 19, 21
- [SD13] STELLMACH S., DACHSELT R.: Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2013), ACM, pp. 285–294. doi:10.1145/2470654.2470695. 15, 22

- [SFB94] STONE M. C., FISHKIN K. P., BIER E. A.: The Movable Filter as a User Interface Tool. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (1994), ACM, pp. 306–312. doi:10.1145/191666.191774. 2, 22
- [SFR01] STOEVS S. L., FEURER M., RUCKABERLE M.: Exploring the Past: A Toolset for Visualization of Historical Events in Virtual Environments. In *Proc. of ACM Symposium on Virtual Reality Software and Technology (VRST)* (2001), ACM, pp. 63–70. doi:10.1145/505008.505020. 12, 21
- [SG07] SHOEMAKER G., GUTWIN C.: Supporting Multi-Point Interaction in Visual Workspaces. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2007), ACM, pp. 999–1008. doi:10.1145/1240624.1240777. 21
- [SHER99] SHAW C. D., HALL J. A., EBERT D. S., ROBERTS D. A.: Interactive Lens Visualization Techniques. In *Proc. of IEEE Visualization Conference (Vis)* (1999), IEEE Computer Society, pp. 155–160. doi:10.1109/VISUAL.1999.809882. 12, 21
- [Shn83] SHNEIDERMAN B.: Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer* 16, 8 (1983), 57–69. doi:10.1109/MC.1983.1654471. 7, 22
- [Shn96] SHNEIDERMAN B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. of IEEE Symposium on Visual Languages (VL)* (1996), IEEE Computer Society, pp. 336–343. 3, 8, 22
- [SND10] SCHMIDT S., NACENTA M. A., DACHSELT R., CARPENDALE S.: A Set of Multi-Touch Graph Interaction Techniques. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2010), ACM, pp. 113–116. doi:10.1145/1936652.1936673. 9, 12, 14, 21
- [SP13] SEDIG K., PARSONS P.: Interaction Design for Complex Cognitive Activities with Visual Representations: A Pattern-Based Approach. *AIS Transactions on Human-Computer Interaction* 5, 2 (2013), 84–133. 17
- [SS14] SADANA R., STASKO J.: Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2014), ACM, pp. 265–272. doi:10.1145/2598153.2598163. 14, 22
- [SSD09] SPINDLER M., STELLMACH S., DACHSELT R.: PaperLens: Advanced Magic Lens Interaction Above the Tabletop. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2009), ACM, pp. 69–76. doi:10.1145/1731903.1731920. 7, 14, 15, 22
- [SSND11] STELLMACH S., STOEVS S., NÜRNBERGER A., DACHSELT R.: Designing Gaze-supported Multimodal Interactions for the Exploration of Large Image Collections. In *Proc. of 1st Conference on Novel Gaze-Controlled Applications* (2011), ACM, pp. 1:1–1:8. doi:10.1145/1983302.1983303. 15, 22
- [SSRG11] SÖRÖS G., SEICHTER H., RAUTEK P., GRÖLLER E.: Augmented Visualization with Natural Feature Tracking. In *Proc. of International Conference on Mobile and Ubiquitous Multimedia (MUM)* (2011), ACM, pp. 4–12. doi:10.1145/2107596.2107597. 15, 22
- [SSS02] STOEVS S. L., SCHMALSTIEG D., STRASSER W.: The Through-the-Lens Metaphor: Taxonomy and Application. In *Proc. of IEEE Virtual Reality Conference (VR)* (2002), IEEE Computer Society, pp. 285–286. doi:10.1109/VR.2002.996541. 22
- [SSTR93] SARKAR M., SNIBBE S. S., TVERSKY O. J., REISS S. P.: Stretching the Rubber Sheet: A Metaphor for Viewing Large Layouts on Small Screens. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (1993), ACM, pp. 81–91. doi:10.1145/168642.168650. 22
- [STSD10] SPINDLER M., TOMINSKI C., SCHUMANN H., DACHSELT R.: Tangible Views for Information Visualization. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2010), ACM, pp. 157–166. doi:10.1145/1936652.1936684. 10, 12, 13, 15, 21
- [TAS09] TOMINSKI C., ABELLO J., SCHUMANN H.: CGV – An Interactive Graph Visualization System. *Computers & Graphics* 33, 6 (2009), 660–678. doi:10.1016/j.cag.2009.06.002. 4, 5, 6, 7, 8, 10, 12, 21
- [TAVHS06] TOMINSKI C., ABELLO J., VAN HAM F., SCHUMANN H.: Fisheye Tree Views and Lenses for Graph Visualization. In *Proc. of International Conference Information Visualisation (IV)* (2006), IEEE Computer Society, pp. 17–24. doi:10.1109/IV.2006.54. 4, 6, 12, 13, 17, 21
- [TFS08] THIEDE C., FUCHS G., SCHUMANN H.: Smart Lenses. In *Proc. of Smart Graphics (SG)* (2008), Springer, pp. 178–189. doi:10.1007/978-3-540-85412-8\_16. 4, 6, 8, 21
- [TGBD08] TRAPP M., GLANDER T., BUCHHOLZ H., DOLNER J.: 3D Generalization Lenses for Interactive Focus + Context Visualization of Virtual City Models. In *Proc. of International Conference Information Visualisation (IV)* (2008), IEEE Computer Society, pp. 356–361. doi:10.1109/IV.2008.18. 12, 21
- [TGK\*14] TOMINSKI C., GLADISCH S., KISTER U., DACHSELT R., SCHUMANN H.: A Survey on Interactive Lenses in Visualization. In *EuroVis State-of-the-Art Reports* (2014), Eurographics Association, pp. 43–62. doi:10.2312/eurovisstar.20141172. 3
- [Tom11] TOMINSKI C.: Event-Based Concepts for User-Driven Visualization. *Information Visualization* 10, 1 (2011), 65–81. doi:10.1057/ivs.2009.32. 7, 22
- [TPH14] TROIANO G. M., PEDERSEN E. W., HORNBAEK K.: User-defined gestures for elastic, deformable displays. In *Proc. of International Working Conference on Advanced Visual Interfaces (AVI)* (2014), ACM, pp. 1–8. doi:10.1145/2598153.2598184. 22
- [TSAA12] TOMINSKI C., SCHUMANN H., ANDRIENKO G., ANDRIENKO N.: Stacking-Based Visualization of Trajectory Attribute Data. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2565–2574. doi:10.1109/TVCG.2012.265. 5, 12, 21
- [UvK13] UKROP J., ČÍKOVÁ Z., KAPEC P.: Visual Access to Graph Content Using Magic Lenses and Filtering. In *Proc. of Spring Conference on Computer Graphics (SCCG)* (2013), ACM, pp. 23–30. doi:10.1145/2448531.2448534. 12, 21
- [VCWP96] VIEGA J., CONWAY M., WILLIAMS G. H., PAUSCH R. F.: 3D Magic Lenses. In *Proc. of ACM Symposium on User Interface Software and Technology (UIST)* (1996), ACM, pp. 51–58. doi:10.1145/237091.237098. 12, 21
- [vHvW04] VAN HAM F., VAN WIJK J.: Interactive Visualization of Small World Graphs. In *Proc. of IEEE Symposium Information Visualization (InfoVis)* (2004), IEEE Computer Society, pp. 199–206. doi:10.1109/INFVIS.2004.43. 10, 11, 12, 21
- [VKG04] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-Driven Volume Rendering. In *Proc. of IEEE Visualization Conference (Vis)* (2004), IEEE Computer Society, pp. 139–145. doi:10.1109/VISUAL.2004.48. 21
- [VTS\*09] VOIDA S., TOBIASZ M., STROMER J., ISENBERG P., CARPENDALE S.: Getting Practical with Interactive Tabletop

- Displays: Designing for Dense Data, “Fat Fingers,” Diverse Interactions, and Face-to-face Collaboration. In *Proc. of International Conference on Interactive Tabletops and Surfaces (ITS)* (2009), ACM, pp. 109–116. doi:10.1145/1731903.1731926. 7, 14, 22
- [WCG03] WONG N., CARPENDALE M. S. T., GREENBERG S.: EdgeLens: An Interactive Method for Managing Edge Congestion in Graphs. In *Proc. of IEEE Symposium Information Visualization (InfoVis)* (2003), IEEE Computer Society, pp. 51–58. doi:10.1109/INFVIS.2003.1249008. 9, 12, 13, 21
- [WEE03] WEISKOPF D., ENGEL K., ERTL T.: Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (2003), 298–312. doi:10.1109/TVCG.2003.1207438. 21
- [WGK10] WARD M., GRINSTEIN G., KEIM D.: *Interactive Data Visualization: Foundations, Techniques, and Applications*. A K Peters, 2010. 1, 22
- [WKS07] WALDNER M., KALKUSCH M., SCHMALSTIEG D.: Optical Magic Lenses and Polarization-Based Interaction Techniques. In *Proc. of Workshop on Immersive Projection Technology, Eurographics Workshop on Virtual Environments (IPT/EGVE)* (2007), Eurographics Association, pp. 61–68. doi:10.2312/EGVE/IPT\_EGVE2007/061-068. 11, 22
- [WZMK05] WANG L., ZHAO Y., MUELLER K., KAUFMAN A. E.: The Magic Volume Lens: An Interactive Focus+Context Technique for Volume Rendering. In *Proc. of IEEE Visualization Conference (Vis)* (2005), IEEE Computer Society, p. 47. doi:10.1109/VISUAL.2005.1532818. 9, 12, 21
- [YCB05] YANG Y., CHEN J., BEHESHTI M.: Nonlinear Perspective Projections and Magic Lenses: 3D View Deformation. *Computer Graphics and Applications* 25, 1 (2005), 76–84. doi:10.1109/MCG.2005.29. 22
- [YKSJ07] YI J. S., KANG Y. A., STASKO J., JACKO J.: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1224–1231. doi:10.1109/TVCG.2007.70515. 3, 10, 17, 22
- [ZCB11] ZHAO J., CHEVALIER F., BALAKRISHNAN R.: KronoMiner: Using Multi-foci Navigation for the Visual Exploration of Time-series Data. In *Proc. of SIGCHI Conference Human Factors in Computing Systems (CHI)* (2011), ACM, pp. 1737–1746. doi:10.1145/1978942.1979195. 12, 21
- [ZCPB11] ZHAO J., CHEVALIER F., PIETRIGA E., BALAKRISHNAN R.: Exploratory Analysis of Time-Series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2422–2431. doi:10.1109/TVCG.2011.195. 4, 5, 8, 9, 12, 21
- [ZZG\*12] ZHAO X., ZENG W., GU X. D., KAUFMAN A. E., XU W., MUELLER K.: Conformal Magnifier: A Focus+Context Technique with Local Shape Preservation. *IEEE Transactions on Visualization and Computer Graphics* 18, 11 (2012), 1928–1941. doi:10.1109/TVCG.2012.70. 21