SchemeLens: A Content-Aware Vector-Based Fisheye Technique for Navigating Large Systems Diagrams

Aurélie Cohé, Bastien Liutkus, Gilles Bailly, James Eagan, Eric Lecolinet
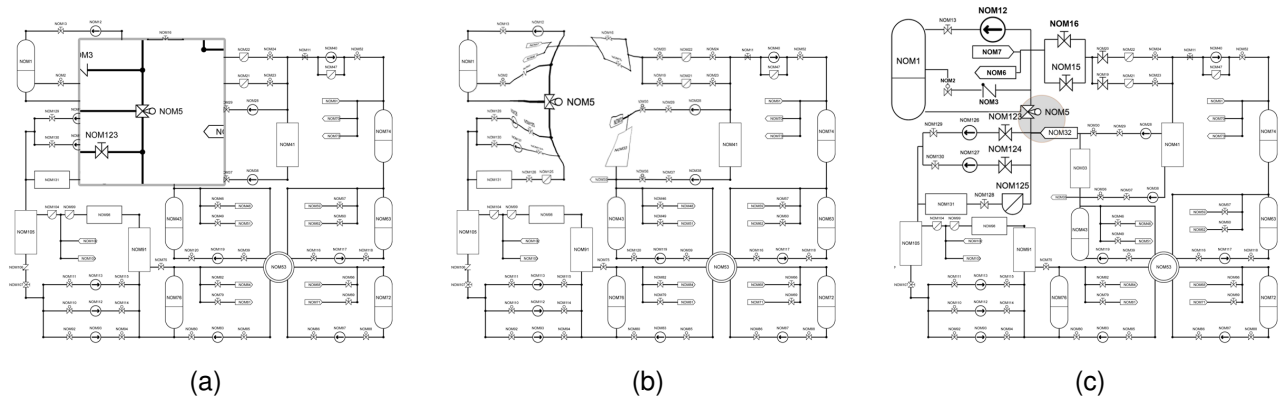
Fig. 1. Three focus+context visualizations of a hydraulic system schematic: (a) a magnifying lens, (b) a fisheye lens, and (c) SchemeLens.

Abstract— System schematics, such as those used for electrical or hydraulic systems, can be large and complex. Fisheye techniques can help navigate such large documents by maintaining the context around a focus region, but the distortion introduced by traditional fisheye techniques can impair the readability of the diagram. We present SchemeLens, a vector-based, topology-aware fisheye technique which aims to maintain the readability of the diagram. Vector-based scaling reduces distortion to components, but distorts layout. We present several strategies to reduce this distortion by using the structure of the topology, including orthogonality and alignment, and a model of user intention to foster smooth and predictable navigation. We evaluate this approach through two user studies: Results show that (1) SchemeLens is 16–27% faster than both round and rectangular flat-top fisheye lenses at finding and identifying a target along one or several paths in a network diagram; (2) augmenting SchemeLens with a model of user intentions aids in learning the network topology.

Index Terms—Fisheye; vector-scaling; content-aware; network schematics; interactive zoom; navigation; information visualization.

---◆---

1 INTRODUCTION

Focus+context interfaces [20] provide a global view of a document while showing a detailed focus region within its context. These approaches face a tradeoff between making more space for the focus region by either occluding information (e.g., magnification lenses, Figure 1a) or by distorting the representation (e.g., Fisheye lenses, Figure 1b). While Fisheye lenses eliminate occlusion, they introduce a distortion that can diminish the readability and comprehensibility of the content. When applied to a network diagram, this distortion causes edges to bend and to bunch, and can cause empty space to unnecessarily grow in importance (Figure 1b). Various approaches have been successful at reducing and constraining this distortion (e.g., [5, 6, 7, 33]), but ultimately they can still interfere with the reading of the schematic.

- Aurélie Cohé is a postdoctoral researcher at Télécom ParisTech. E-mail: aurelie.cohe@telecom-paristech.fr.
- Bastien Liutkus is a research engineer at Télécom ParisTech. E-mail: bastien.liutkus@telecom-paristech.fr.
- Gilles Bailly is a researcher at the CNRS LTCI & Télécom ParisTech. E-mail: gilles.bailly@telecom-paristech.fr.
- James Eagan is an assistant professor at Télécom ParisTech & the CNRS LTCI. E-mail: james.eagan@telecom-paristech.fr.
- Eric Lecolinet is an associate professor at Télécom ParisTech & the CNRS LTCI. E-mail: eric.lecolinet@telecom-paristech.fr.

In this paper, we present SchemeLens, a content-aware, vector-based fisheye technique for the exploration of large and complex network diagrams, such as electrical, mechanical, hydraulic diagrams, or biological pathways. In these diagrams, the reader needs to maintain an overview of the overall structure of the network while being able to read the labels of individual components of interest. SchemeLens has several advantages. (1) It uses vectorial scaling of the focus region (Figure 1c) to preserve the form of individual components and to avoid scaling up empty space. Additionally, (2) it uses the topology of the network diagram to scale up components not only based on their geometric proximity to the cursor but also by their topological proximity.

A naïve vector-based scaling of components would result in overlapping components. By being aware of the network topology, (3) components can shift their positions to maintain a topologically equivalent drawing that moves and re-scales non-focus elements to make room for (scaled) focus components. Aesthetics rules can enforce orthogonality and alignment. The combination of these properties effectively causes the shape of the lens to adapt itself to the specific network components and to their topology.

Interactive fisheye techniques must further provide for smoothly changing the focus. Taking into account euclidean distances in the topology-based degree-of-interest (DOI) function for nodes helps to smooth out node resizing. (4) A force-based attraction of nodes to their initial positions acts as a sort of inertial dampener. Our primary contributions are:

- The design & implementation of a vector-based, content-aware fisheye lens for network diagrams,

- A space-making strategy that prioritizes structurally-similar layouts (e.g., alignment and symmetry in redundant components),
- Two user studies showing that (1) SchemeLens is 16–27% faster than both round and rectangular flat-top fisheye lenses for path-following tasks and (2) augmenting SchemeLens with a lens that adapts itself to user intentions aids in learning the network topology.

2 MOTIVATION

In order to inform the design of SchemeLens, we conducted a series of in-situ interviews with three engineers who routinely work with large diagrams for complex, safety-critical systems. In particular, we wanted to understand what kinds of data the engineers worked with, what form it took, and what kinds of analyses they performed when working with the data.

Many of these systems have evolved over the course of decades and are thus represented in a mixture of paper and digital formats. Furthermore, the systems are complex and are frequently broken down into multiple sub-diagrams. Even when printed out on A3 paper (about $11.7'' \times 16.5''$), a single diagram typically only shows a part of the system. Engineers thus navigate frequently between printed documents. Electronic versions are also available, but the current tools provide only basic pan & zoom capabilities. As such, engineers would often first work on paper, which provides a larger overview, before transferring their work to the electronic versions. For safety-critical systems, it is especially important to avoid mistakes induced by panning past a critical valve or confusing a redundant system.

One common task performed by engineers in a variety of contexts is to follow a path in a diagram, such as when defining procedures for isolating a particular circuit for maintenance. Currently, engineers will frequently use a highlighter on a printout of the schematic in order to work out a particular path. Additionally, operators will often use these schematics when learning or rehearsing a new procedure in order to better identify and understand the components involved.

From these interviews, it became clear that a real need exists in the navigation of such large network diagrams. As such, we defined several high-level goals for such a visualization:

Goal 1: Maintain readability of nodes of interest.
Goal 2: Minimize distortion to the structure of the diagram.
Goal 3: Maintain a stable structure during interaction.

The first design goal is to maintain the readability of nodes of interest while showing the full context. As such, nodes in the focus area should be clearly and unambiguously legible. Any distortion to the presentation should not lead to the potential confusion of two similar kinds of nodes. The second of these goals is to minimize distortion to the structure of the graph in both the focus and the context regions. Users should be able to maintain a clear mental model of where they are in the diagram [12, 21]. In particular, orthogonal connections should remain orthogonal and symmetries in redundant components should be clear. Users should be able to reasonably identify redundant components and which one is currently being viewed. Reducing distortion emphasizes readability (by guiding attention), promotes understanding [8], and can reduce cognitive load. Finally, the schematic should remain stable during navigation. Small changes to the focus region should result in smooth, continuous changes to the overall representation. Users should not have to spend much cognitive effort to reassimilate the configuration of the diagram when interactively changing the focus.

3 RELATED WORK

Fisheye views [9, 14] (Figure 1b), which are probably the most well-known example of focus+context techniques, distort the space to present a zoomed region that smoothly integrates with the surrounding context. Various curves and shading models have been proposed to improve the readability of the zoomed and distorted regions [6, 7, 33]. Nonetheless, the warping necessary to create a smooth transition can hinder readability, so much that "undistortion lenses" [5] have been proposed to let users interactively remove distortion from certain regions.

While the name "fisheye" often evokes the spherical distortion associated with a fish's round eye, other distortions are possible [6, 9, 33]. One such example is Furnas's code editor that uses dynamic code folding to show only relevant regions of source code [9] based on their semantic importance and distance from the cursor. In this generalized model, the degree-of-interest (DOI) of objects (such as lines of code) is a function of their inherent importance and their distance from a focus. For interactive visualizations, it is important that this DOI function be continuous so as to ensure smooth transitions and avoid disorienting the user when changing the focus.

Topology-aware Fisheyes. Generalized fisheyes often rely on an understanding of the underlying content in order to calculate the intrinsic value of an object. For example, Sarkar & Brown [25] use knowledge of graph structures to apply fisheye distortion to graph visualizations in a way that maintains the readability of the graph. However, topologically-aware fisheye techniques may introduce discontinuities, such as when navigating from one side of a node to another. Scheme-Lens addresses this problem by blending two components: 1) a component taking into account the euclidean distance to avoid discontinuities when nodes are resized, and 2) a spring-based model which acts as an inertial dampener when the nodes are displaced.

Perhaps closest in spirit to SchemeLens is work by Kim et al. and Woo et al. [13, 32], who studied the interactive navigation of technical diagrams. In this work, they compare a rectangular fisheye lens [22] and a magnification lens [3] for navigating schematics. They find a significant accuracy benefit to the fisheye lens and argue that the magnifying distortion enables users to see more contextual information and to preserve the views of adjacent regions. Nonetheless, such rectangular fisheye lenses use a rubber-sheet model [26], which despite its readability advantages, still causes components outside the focus region to be stretched and deformed, potentially hindering symbol recognition.

Adaptive Lenses. By using an understanding of the underlying content, the fisheye can adapt its DOI function or its shape based on the particular focus region. For example, Gansner et al. use a semantic zooming-based approach to generate simplified subgraphs for low-interest portions of very large graphs [10]. JellyLens [19], on the other hand, adapts its shape to match features of the underlying focus. It uses a visual saliency model to infer the structure of the content from its pixels, and adapts the size and the shape of the lens such that the transition region of maximum distortion occurs in low-relevance regions. Because it uses a visual model of content, edges and components in transition regions are still warped. By using a topologically-aware vector scaling approach, the shape of the SchemeLens lens effectively adapts itself to the individual components in the diagram.

In their survey of the literature on interactive lenses, Tominski et al. [29] emphasize the fact that "the fundamental problem is how to provide the user a quick and easy way to: position the lens, work through the lens, and (possibly) parameterize the lens." They also call out the need to flexibly combine lenses to create new lens functions on the fly. SchemeLens addresses these requirements by providing a combination of lenses that the user can activate or deactivate (such as by toggling selectable geometric and topological lens rules).

Navigation techniques. Content-aware methods can facilitate navigation within a document. For example, Link Sliding and Bring & Go [17] help a user to follow the structure of a graph, even when zoomed into a scale where no overview is available. For example, Bartram et al. and Shaffer et al.'s continuous zoom [2, 27] uses a model of an underlying hierarchy. When the user zooms in on a focus region, the necessary space comes first from empty regions, then by shrinking hierarchically-distant components. SchemeLens also uses a model of underlying content beyond simple hierarchical relations to restrict objects' movements to maintain coherent patterns, such as by maintaining alignment in a row of components or in redundant structures.

Target-aware techniques can also use a model of the user's intent to further guide the user along a specific path, such as shown by Alvina et al. or by Wang et al. for map guidance [1, 30]. Beyond simply navigating within a graph, the model of the underlying content can help the user to reconfigure the representation in ways that make sense. In the same way, SchemeLens takes user intent into account to favor nodes in the direction of navigation.

Perception & Readability. Spatial arrangements are common, and spatial memory can play an important role in information search tasks [28]. Spatial memory is only reliable, however, when objects maintain stable spatial arrangements. As such, we aim to constrain object displacements to predictable locations. Our goal is that such displacements avoid interfering with the user's spatial model of the graph. Wong et al. [31] have further observed that, when displacing edges to make dense edge bundles easier to follow, nodes should maintain a consistent position. As such, it is important to constrain distortions so as to maintain readability.

One approach we follow is to avoid distorting the form of objects. In their exploration of distortion, Carpendale et al. [6] observed that distorted views of familiar layouts, including text, can be uncomfortable. Perspective-based distortions can help reduce this discomfort [23], but are still more difficult to read than undistorted text. Technical diagrams make frequent use of learned symbols whose distortion would hinder their recognition and readability. As such, in our design, we restrict our distortions to scaling and translation. Furthermore, we maintain horizontal and vertical alignment when applying fisheye distortions so as to maintain a coherence with the orthogonality of un-zoomed features. This approach should also help users comprehend diagrams because visual features, such as orthogonality, collinearity, and symmetry, have been shown to aid accurate recall of network diagrams [15].

4 THE SCHEMELENS TECHNIQUE

In order to facilitate the navigation and exploration of large system diagrams, SchemeLens combines a vectorial fisheye zoom with a model of the network topology. To achieve this goal while maintaining readability in the focus region and avoiding distortions that hinder readability outside the focus region, we combine several components (see Figure 2), in particular:

- A *magnifying lens* which magnifies individual nodes located near the cursor.
- A *topological lens* which propagates that magnification outward to neighboring nodes based on their geodesic distance.
- An *aesthetic model* which preserves layout properties of the original diagram such as node alignment, redundant component alignment, or orthogonality.
- A *user intent* model to dynamically favor nodes based on the user's current actions.

We use the term "distortion" to refer to operations that alter the initial representation of the diagram, such as translation, rotation, scaling, stretching, or warping. Some distortions can help to increase the readability of the diagram, while others can hinder it. The readability of text, symbols and diagram structure refers to the ease with which the user can read text, recognize shapes and symbols, and generally get a good overview of the structure (e.g., the user is able to know that two components are linked, or that two paths are parallel).

The following subsections describe how SchemeLens attempts to maximize the readability of the nodes and connections of interest (the focus area) while minimizing distortion to the diagram. The last subsection provides details on the implementation.

4.1 Focus Area And Readability

4.1.1 Magnification Lens

As with other fisheye techniques, SchemeLens magnifies elements that are close to the cursor. It uses a vectorial zoom to maintain the form of nodes. Nodes within a certain radius of the cursor are magnified, with the magnification falling off to zero at the edge of the lens. The magnification of a given node depends on the distance from its barycenter to the center of the lens.

4.1.2 Topological Lens

In user interviews, participants highlighted the need to be able to follow a path node-by-node, by reading off a sequence. Such path following is a common activity highlighted in the literature [32]. To help this task, we propagate the magnification of nodes near the focus to its topological neighbors based on their geodesic distance, with farther neighbors receiving less of a magnification. This mechanism helps to call out connected nodes that may be geographically far apart.

4.1.3 Adaptative Focus Area

A key property of SchemeLens is that the focus area is the union of the area covered by the *magnification lens* and of the area covered by the *topological lens*. In other words, a node can be magnified even if it is not in the vicinity of the cursor. The focus area can thus be seen as the result of a content-aware adaptive lens akin to that of JellyLens [19]. Although we use a circular selection area to determine the magnification lens area, individual nodes are scaled uniformly, as if the shape of the lens had perfectly matched the outline of the affected components. This design has several advantages related to readability and content-aware scaling.

Readability. When a node is magnified, this operation is performed uniformly on the whole node. For example, textual labels are magnified just by changing the font size, maintaining a uniform scale across the label. Not only does this make text and graphics crisper than raster scaling, but it also maintains readability, which is hindered by distortion such as warping [6].

Merely using vector scaling to magnify nodes is insufficient as nearby nodes may overlap one another. To avoid this problem, SchemeLens moves nodes to avoid superimposition (design goal 1) but aims at minimizing these displacements (design goal 2). To balance these goals, we use a spring-based model: a repulsion force between nodes avoids overlaps while an attractive force encourages nodes not to stray too far from their original positions. In this way, SchemeLens uses a vector-based zoom to uniformly magnify nodes while taking into account their initial positions (for displacements).

Content-Aware Scaling. One challenge when performing a vectorial zoom rather than a raster zoom is that a given change to the scale of a large object can carry a disproportionate visual weight to the same scale change of a small object. Thus, when magnifying nodes, we apply an inverse exponential function to modulate the scaling factor depending on the node's intrinsic size (see Equation 2). This prevents large nodes from becoming disproportionately large once magnified.

4.1.4 Size of the focus area

In order to be useful, the magnification effect should be limited to a "reasonable" number of nodes. Of course, what "reasonable" means in this context is a difficult question because it may both depend on user preferences and on various factors such as the local topology (e.g., the number of branches) and the local density (e.g., the proximity of nodes to each other) of the diagram. Since visual throughput is limited and human beings can only deal simultaneously with a small number of items [16], one solution would be to limit the number of magnified nodes (e.g., to seven nodes [16]). This solution would be problematic when the focus is near a branch: very few nodes would thus be magnified on each branch, limiting the utility of the topological lens.

We use an alternative approach of setting a threshold on the geodesic distance of magnified nodes, which is independent of the number of branches. Some topologies, however, may have fairly shallow spanning trees across large parts of the topology. We currently use a default threshold of three and let users control this threshold or the size of the focus area using a combination of the mouse wheel and modifier keys.
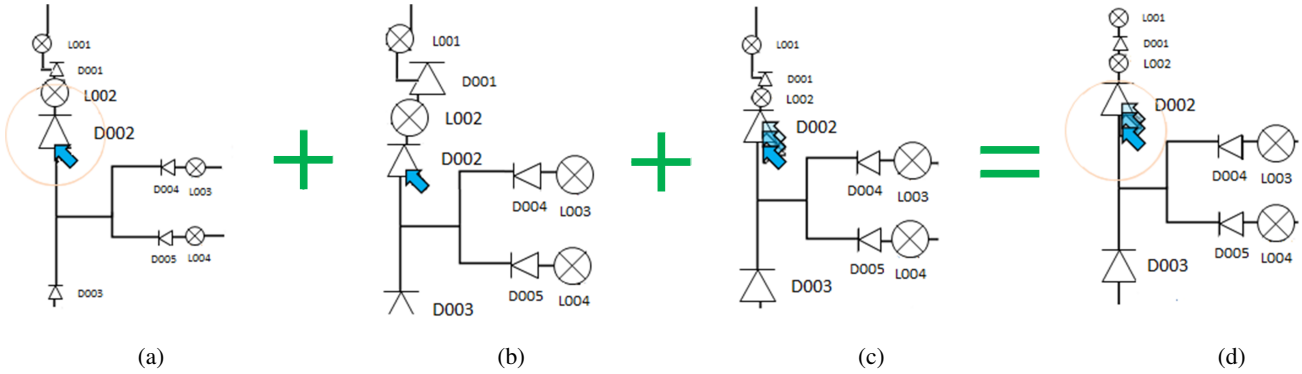
(a)   (b)   (c)   (d)

Fig. 2. (a) Magnification lens: elements that are close to the cursor in the focus area represented by the red circle are magnified. (b) Topological lens: the element under the cursor is magnified and this magnification is propagated along the paths starting from this node. (c) Capturing user intent: When moving the cursor, only the nodes in the direction of these movements along the path are magnified. (d) Schemelens is a combination of these methods.

4.2 Context Area and Distortion

Magnifying elements in the focus area increases their readability, but it can also introduce distortion by causing other elements to move around. To help preserve spatially-similar layouts and to maintain the reader's spatial model of the diagram, we use several heuristics aimed at maintaining spatial stability, aesthetic properties, and temporal stability.

4.2.1 Aesthetic rules

SchemeLens enforces several *aesthetic rules* based on the structure and layout of the diagram. System schematics generally consist of orthogonal lines. SchemeLens creates a three-level hierarchical structure that respectively stores the nodes of the diagram, the straight lines that contain these nodes and the sub-paths that contain these lines (a sub-path being a part of the diagram with no crossing point). This structure preserves the following rules:

- *Alignment.* Nodes on a same line remain aligned along the axis of the line (see Figure 3). Nodes on different lines within a local loop also remain aligned (see Figure 4). Intersections (i. e., crossings between paths) that are aligned remain aligned (see Figure 5). This last rule is the same as alignment of nodes if we consider crossings as nodes.
- *Orthogonality.* The orthogonality of lines is preserved.
- *Edge crossing.* No new edge crossings are introduced.
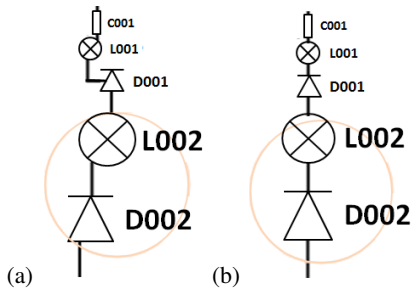

(a)   (b)

Fig. 3. Alignment of nodes on the same path: (a) Node alignment is not taken into account. (b) Node alignment along a path is preserved.

These rules help to preserve the key visual properties of system diagrams. They are implemented as constraints that control the displacement of nodes and could be changed to enforce other properties (e.g., for different kinds of diagrams). This constraint system and the underlying hierarchical structure could be enriched to take into account more complex topological structures.
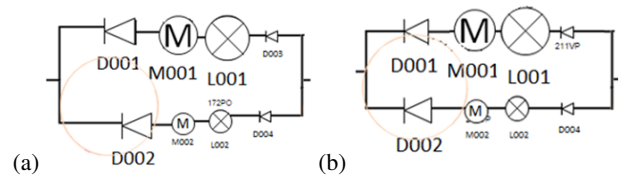

(a)   (b)

Fig. 4. Alignment of nodes on the same loop: (a) Nodes are moved independently (b) Alignment of nodes within a loop is preserved.
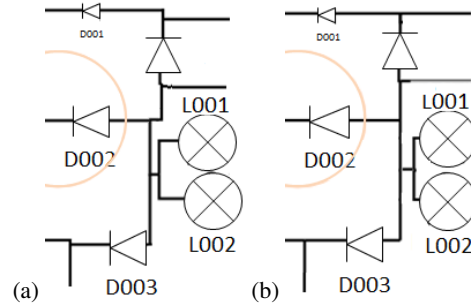

(a)   (b)

Fig. 5. Alignment of intersections: (a) Intersection alignment is not taken into account. (b) Edge intersections remain aligned.

4.2.2 Node displacement

In order to make room for the focus area, nodes in the focus area need to be moved. A repulsive force is used to push nodes into empty areas and to avoid overlaps between nodes. Moreover, we add an attractive force to each node's original position to reduce spatial distortion.

Furthermore, the vectorial zooming used for the focus area does not magnify empty space, thus decreasing the amount of space needed from the context area. Instead, magnified nodes grow first into the empty space in the focus area (see Figure 1c). This approach also helps to avoid the vortex effect of maximum distortion around the edge of the lens [7] because it exploits empty space as a kind of buffer, hence increasing continuity between the focus and context areas.

4.2.3 Spatial and Temporal Stability

One key challenge with SchemeLens is to maintain a high level of spatial stability: without appropriate precautions, small changes to the focus position could introduce radical changes to the position or size of elements. Navigating could also introduce a disconcerting yo-yo effect where changes are smooth but the geometrical properties of the elements appear unstable with a seesaw motion, such as when moving the focus along a path between two nodes. We used two different

strategies to maintain the stability of the position and the size of elements:

Position. The spring-based model not only helps to maintain a stable layout when moving nodes to make room for the focus, but it also acts as a sort of inertial dampener to avoid temporal discontinuities and to provide smooth interaction.

Size. Size instability is more difficult to predict and happens when users are following a path. It results from two contradictory rules because of the simultaneous use of two different lenses (the magnification lens and the topological lens). Nodes that are topologically close to the cursor are magnified, but nodes that are at the edge of the magnification lens are not. As a result, (1) the size of a node could decrease even as the user approaches it with the cursor, and (2) the scale of a node could change abruptly at certain boundaries.

To avoid this problem, our algorithm ensures that the size of a node cannot decrease when the user follows a path in direction of this node. A side effect is that it prevents nodes from being abruptly scaled down when entering in the magnifying lens.

4.2.4 Capturing User Intent

Optionally, the system can take *user intent* into account to favor nodes in the direction of navigation (see Figure 2c). When the user moves the cursor to follow a path, the system shifts the focus toward nodes in the direction of this movement. Thus, when the cursor moves along a path, only the nodes in the direction of the cursor along the path are magnified. If it does not follow a path (e.g., the direction of the cursor is perpendicular to the closest path or the cursor is far from all paths), this behavior is not applied. In the current implementation, and based on our pilot studies, we set a 10-pixel tolerance for path following. A small "dead zone" helps to avoid involuntary actions.

The part of the diagram which is magnified could also be colored to increase user attention. However, this option was not retained in the user experiment to avoid introducing a new factor that would have made it difficult to compare SchemeLens with other techniques. Moreover, such coloring could interfere with other color-based techniques, such as highlighting a specific path.

4.3 Implementation

Magnification Lens. Nodes in the magnification lens are scaled based on their DOI and a computed scale factor (described below). We first calculate the DOI (DOI_n^{ml}) of the nodes located inside the magnification lens as follows: for a given node n:

$$DOI_n^{ml}(P_c, P_n) = doi_{max} - \frac{(doi_{max} - doi_{min}) \times d_e(P_c, P_n)}{r} \quad (1)$$

where r is the radius of the circular magnifying lens; $d_e(P_c, P_n)$ is the euclidean distance between the cursor, c, at the center of the lens, and the center of the initial position of node n; and doi_{min} and doi_{max} control the minimum and maximum node sizes inside the lens.

The scaling factor (SF_n) of each node n is then given by:

$$SF_n(P_c, P_n, S_n) = 1 + (DOI_n(P_c, P_n) - 1) \times \exp\left(-\frac{S_n}{\alpha}\right) \quad (2)$$

where S_n is the surface of node n and α an attenuation constant. The exponential function serves to reduce the magnification of large elements to avoid them taking unnecessary space. Because of their large size, scaling them up directly by their DOI would result in them carrying a disproportionate perceived weight. The α constant needs to be set according to the mean size of the nodes of the diagram (a value of 5000 was used in our experiments). The value of the scaling factor is always greater than 1.0.

Topological Lens. We define n_c to be the nearest node (in euclidean space) to the cursor. It is thus the most magnified node. The DOI

(DOI_n^{tl}) of a node n outside the magnification lens (but connected to a node inside the lens with a sufficiently small geodesic distance, as explained above) is calculated as follows:

$$DOI_n^{tl}(n, P_c) = \frac{DOI_n^{ml}(NML(n)) - doi_{min}}{d_g(n, n_c) + continVar(n_c, c)} + doi_{min} \quad (3)$$

where $NML(n)$ is a function returning the nearest node from n (in the sense of the geodesic distance) located in the magnifying lens (see Figure 6) and $d_g(n)$ the geodesic distance from n to n_c. doi_{min} is the minimal DOI (see Equation 1). Finally, $continVar()$ (described below) serves to maintain continuity during interaction and is always between 0 and 1.

We note that DOI_n^{tl} for a node n is between 1.0 and the DOI of the the topologically nearest node ($NML(n)$) in the magnifying lens. This ensures that nodes in the topological lens have a smaller DOI than nodes in the magnification lens. The node n is then scaled using the scale factor (Equation 2).
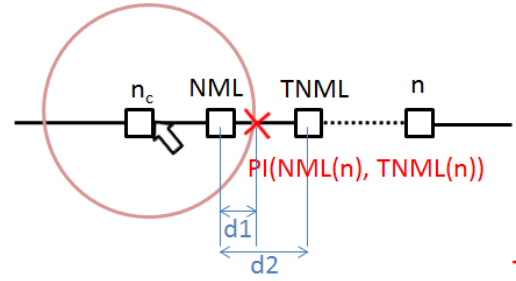

Fig. 6. Illustration indicating the different nodes and point used in the formula of the function *continVar*.

Node displacement. Our spring-based technique is implemented with Javascript and uses the d3js library. The force algorithm uses a force-directed graph layout using position Verlet integration and relies on a quadtree to accelerate charge interaction using the Barnes-Hut approximation. In our implementation, the value of the charge for each node is defined as follows:

$$C_n(n, P_c) = \gamma \times (1 - \exp(\delta \times (SF_n(n, n_c) - 1))) \quad (4)$$

where γ and δ are constants and SF_n the scaling factor of the current node n, calculated from equation 2. After some pretests, we set γ and δ to 3. The charge is thus negative, which results in repulsion between nodes. We also fixed the friction coefficient, which approximates velocity decay, to a value of 0.8.

Spatial and Temporal Stability. The $continVar()$ function (see equation 3) smooths the yo-yo effect described in Section 4.2.3. It takes into account two nodes (see Figure 6): (1) $NML(n)$, the first node from node n in the magnification lens and (2) $TNML(n, n_c)$, the last node from n in the topological lens just after $NML(n)$. These two nodes are neighbors and located on each side of the border of the magnification lens (see Figure 6). We define d_2 as the euclidean distance between these two nodes and d_1 the euclidean distance from NML to the border of the magnification lens. $continVar()$ is thus defined as:

$$continVar(n, n_c) = \frac{d_1}{d_2} \quad (5)$$

5 STUDIES OVERVIEW

We now report on two user studies. The goal of the first user study was to validate our approach and to identify whether SchemeLens was a promising visualization technique for the exploration of system schematics. We thus compared SchemeLens with a round Fisheye lens [9, 14] as a baseline. Results show that SchemeLens is 20–27% faster than a standard fisheye for a path-following task. Encouraged

by these results, we conducted a second study to better understand how the adaptability of SchemeLens impacts performance. We investigated two dimensions of the adaptability of the lens: topology (content awareness) and user intentions. We thus compare four techniques to understand the relative performance of these two properties. Results show (1) SchemeLens is 16%–18% faster than a rectangular top-flat fisheye lens; (2) The combination of topology and user intention adaptability aids in learning the network topology.

5.1 Study 1

5.1.1 Experimental Protocol

Participants and Apparatus. Ten participants from our institution (two female), aged from 23 to 37 (*mean* = 30, σ = 4.7) and with normal or corrected-to-normal vision received candies for their participation. All of them were right-handed. The experiment was conducted on a desktop with a 20-inch screen running at a resolution of 1600 × 900 pixels. The experiment was implemented in Javascript using the libraries D3.js and three.js. The techniques were tested in the Firefox web browser. System schematics were in SVG. To move the cursor, a standard mouse was used and the mouse gain was the same for all participants.

System Schematic. Figure 1c shows the diagram like the one we used in the experiment. This experimental schematic is a vector-based diagram with several aesthetics features such as orthogonality, alignment, etc. It contains 125 paths, 56 components and 58 junctions. The font size (1 millimeter in height) of the contextual view prevented participants from reading text without using the zooming techniques. We also changed the names of elements after each technique to avoid any learning effects. For each trial, some paths were highlighted to guide the exploration. We did not want our participants to explore the entire diagram. Instead, we wanted to control how many nodes and paths they would have to visit. This setup simulated more experienced users who might already have an idea of the subgraph of the diagram containing a faulty (target) node and ensures that the total duration of the experiment was less than 90 minutes.

Techniques. We compared SchemeLens to a traditional fisheye. The chosen fisheye lens relies on a Gaussian drop-off function with a circular shape. We use this implementation as a baseline for alignment with other published studies such as [19, 31]. The two techniques have the same maximum magnification factor (250%) to ensure text readability only in the focus area. The size of the focus area is controlled via the mouse wheel.

Task and Procedure. The experimenter explained the task and how the technique works. Participants could familiarize themselves with each technique during 5 minutes on a training diagram. Once they felt comfortable, they started the experiment.

We derived the task from our interviews with engineers who work with very large system schematics. The task consisted of following one or several paths starting from a given node in order to find a faulty node (e.g., an element that does not work properly and is thus affecting the whole pipeline). Participants first clicked on the starting node indicated in green on the diagram. A stimulus then appeared at the top of screen indicating the name of the target node. Participants explored the subgraph of the diagram to find this node. The trial stopped when participants clicked on the target node. During the experiment, participants could take a break after each trial and between each block.

At the end of the experiment, participants filled in a questionnaire based on a five-point Likert scale from 1 "totally disagree" to 5 "totally agree." We also conducted a semi-structured interview to collect comments about the two techniques.

Task difficulty. In this experiment, we manipulated the difficulty of the task by controlling the number of branches to explore. Participants had to find the target in a path containing 1 (easy), 2 (medium) or 3 (difficult) branches. We hypothesized that the difficulty of the task

(i. e., the number of branches to explore) would impact the relative performance of the SchemeLens and the fisheye techniques.

Design. Each participant tested the round *fisheye* and *SchemeLens*. The order of techniques was counter-balanced between participants. For each technique, participants completed two blocks of 108 trials. The location of the subgraph to explore, the relative position of the target on the path as well as the number of branches to explore appeared in a controlled randomized order to ensure the same difficulty between techniques without offering the opportunity to the participants to "guess" the location of the target. In summary, the design is: 10 users × 2 techniques × 2 blocks × 3 difficulties ({1, 2, 3} number of highlighted branches) × 18 trials = 2160 selections.

5.1.2 Results

Completion Time. Figure 10 shows completion time for each technique and each difficulty. A two-way ANOVA shows a main effect of *Task difficulty* on completion time ($F_{2,18}$ = 78.8, $p \leq 0.001$). A post-hoc test with Bonferroni correction confirms that completion time increases with task difficulty. ANOVA also shows an effect of *Technique* on completion time ($F_{1,9}$ = 9.22, $p \leq 0.05$). A post-hoc test with Bonferroni correction reveals that SchemeLens (7.4s) is significantly faster than fisheye (9.2s). There was no *Technique* × *Difficulty* interaction effect ($F_{2,18}$ = 1.64, $p = 0.24$).
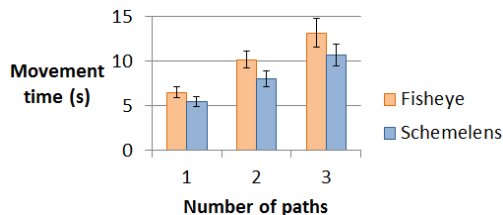

Fig. 7. Movement time (means and standard deviations) for each technique and each number of paths used in the experiment.

Subjective preferences. Figure 8 shows the subjective preferences per technique for each criterion. A Kruskal-Wallis H test showed that there was a statistically significant difference for technique preference ($\chi^2(1)$ = 4.68, $p \leq 0.05$), readability of distant elements ($\chi^2(1)$ = 14.28, $p \leq 0.001$) and memorization ($\chi^1(3)$ = 5.65, $p \leq 0.05$). Indeed, SchemeLens outperforms the Fisheye technique for each of these criteria with a median score of 4.0/5 vs 3.0/5 for technique preference, 2.0/5 vs 5.0/5 for readability of distant elements, and 4.0/5 vs 2.5/5 for memorization. The seven other statements are not significant.
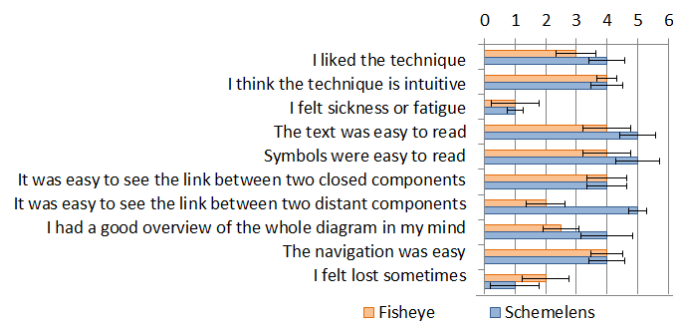

Fig. 8. Subjective marks (medians and confidence intervals) for each statement.
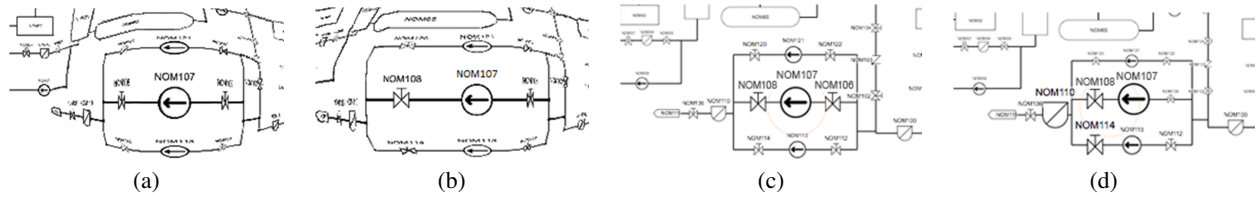
Fig. 9. The techniques used in study 2: (a) Fisheye without user intentions, (b) Fisheye with user intentions, (c) SchemeLens without user intentions, (d) SchemeLens with user intentions.

5.2 User study 2: Evaluation of adaptive lenses

The goal of the second study was to better understand how the adaptability of SchemeLens impacts performance. We investigated two dimensions of the adaptability of the lens:

- Topological adaptation. *"Does the technique adapt the shape of its lens according to the topology of the diagram?"* The presence of this property is a key feature of SchemeLens. Indeed, we hypothesized that taking into account the topology to adapt the lens helps the user in navigating in schematics.
- User Intention adaptation. *"Does the technique adapt the shape of its lens according to the intention of the user?"* We were also interested in this property as it can prevent division of user attention in favoring area in the direction of navigation.

From this design space, we derived four techniques illustrated in Figure 9:

- *Fisheye without UI*. This technique does not adapt its shape and plays the role of baseline. For this study, we used a rectangular fisheye since we compared SchemeLens to a round lens in study 1. Moreover, the lens had a flat-top to help text readability and to reduce possible confounding factors, since SchemeLens relies on a composition of rectangular areas and does not warp text (see Figure 9a).
- *Fisheye with UI*. The same rectangular flat-top fisheye also takes into account the user intention (UI). It increases the size of the lens in the direction of the mouse movement. For instance, the area increases on the left side if the mouse moves toward the left (see Figure 9b).
- *SchemeLens without UI*. Takes the topology of the diagram into account to adapt the shape of its lens (see Figure 9c).
- *SchemeLens with UI*. Also takes into account user intention in addition to the topology of the diagram to adapt the shape of its lens (see Figure 9d).

All four techniques have the same maximum magnification factor (250%). The size of the focus area is controlled via the mouse wheel. We observed that users generally defined this value during the training phase and then did not change it anymore. When some parameters were specific to a technique (e.g., the zoom speed for Fisheye with UI), we conducted pilot studies to optimize the performance of the technique.

5.2.1 Experimental Protocol

Participants and Setup. Sixteen new participants (five female) aged from 21 to 34 ($mean = 26$, $\sigma = 3.1$) from our institution with normal or corrected-to-normal vision received candies for their participation. All of them were right-handed. We reused the same apparatus as in study 1. However, we used four different schematics with the same complexity: 152 ± 3 paths, 95 ± 2 components and 41 ± 1 junctions for each of them. We used different schematics so as to be able to test the impact of the techniques on memorization.

Task and Procedure. We used the same task and procedure as in study 1 except that we introduce a memorization test after each technique. Participants had to draw as many elements (visited and not visited) that they could *correctly* remember from the diagram. Elements

included both nodes and edges that were visited or not during the previous phase. This memorization task was added to investigate the impact of each technique on the implicit learning of the diagram.

Design. Each participant tested all four techniques. The order of techniques was counter-balanced between participants. The location of the subgraph to explore, the relative position of the target as well as the the difficulty of the task (the number of branches to explore) appeared in a controlled randomized order. In summary, the design is: 16 users × 4 techniques × 3 difficulties × 6 trials = 2304 selections.

5.2.2 Results

Completion Time. Figure 10a shows completion time for each technique and each difficulty. A two-way ANOVA shows a main effect of *Task difficulty* on completion time ($F_{2,30} = 40.48$, $p \leq 0.001$). A post-hoc test with Bonferroni correction confirms that completion time increases with task difficulty. ANOVA also shows an effect of *Technique* on completion time ($F_{3,45} = 3.89$, $p \leq 0.05$). A post-hoc test with Bonferroni correction reveals that SchemeLens with UI and without UI (8.2s and 8.0s, respectively) is significantly faster than Fisheye without UI (9.5s). No significant difference was found between Fisheye with UI (9.0s) and the other techniques. There was no *Technique × Difficulty* interaction effect ($F_{6,90} = 1.39$, $p = 0.23$).
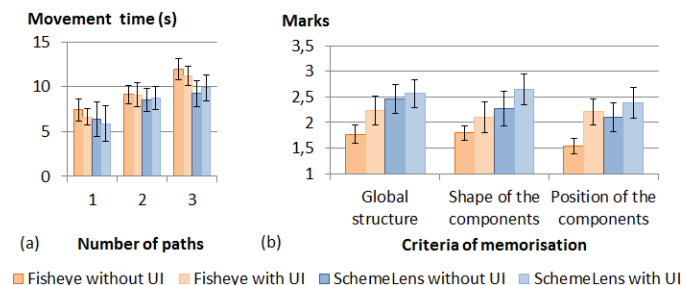

Fig. 10. (a) Movement time (means and standard deviations) for each technique and difficulty level. (b) Memorization (median and standard deviations) on a 5-point Likert scale. 5 is the best.

Memorization. To evaluate memorization, we asked three evaluators to grade using a five-point Likert scale for three criteria for each diagram. The criteria were the memorization about the whole structure of the diagram, about the shape of the components and about the position of the components. When evaluating a diagram, the evaluators knew neither the corresponding technique nor the participant. Figure 10b shows the given grades per criterion and technique. One-way ANOVAs show a main effect of the technique on memorization about the structure of the diagram ($F_{3,45} = 3.70$, $p \leq 0.05$), on memorization about the shape of the components ($F_{3,45} = 3.42$, $p \leq 0.05$) and on memorization about the position of the components ($F_{3,45} = 3.88$, $p \leq 0.05$). Post-hoc tests with Bonferroni reveal that users were significantly better able to memorize the whole structure of the diagram (2.56 vs 1.77), the shape of the components (2.65 vs 1.79) and of the position of the components (2.38 vs 1.54) using SchemeLens with UI vs. the Fisheye without UI.

Subjective preferences. Figure 11 shows the grades that participants gave for statements presented above. A Kruskal-Wallis H test showed that there was a statistically significant difference in the first statement, about technique preference ($\chi^2(3) = 9.47$, $p \leq 0.05$), in the fifth statement, about readability of symbols between the different techniques ($\chi^2(3) = 10.1$, $p \leq 0.05$) and in the seventh statement, about readability of distant elements ($\chi^2(3) = 22.5$, $p \leq 0.001$).
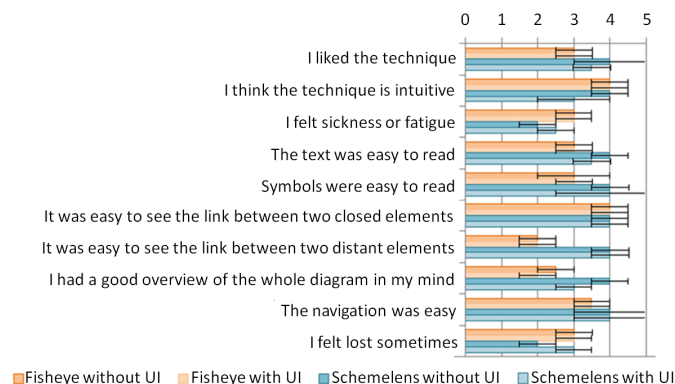

Fig. 11. Subjective grades (medians and confidence intervals) for each statement.

6 DISCUSSION

SchemeLens vs. Fisheye. We compared SchemeLens with two different kinds of fisheye lenses. The first fisheye lens (Study 1), which relies on a round shape lens, is common in literature. The second fisheye lens (Study 2) relies on a rectangular flat-top lens, which should reduce distortion in the focus area. These two fisheyes act as baselines. The results of the studies confirm the effectiveness of our approach: Schemelens is significantly faster than these two fisheye techniques, with an average improvement of 23% for the round shape lens and of 17% for the rectangular flat-top lens. While comparisons with other fisheye techniques, such as [26], would be useful for the sake of generalization, we believe this shows the benefit of considering the topology for the interactive visualization of diagrams.

Topological vs. user intent adaptability. The second study evaluated the influence on time performance of two factors impacting the adaptability of the lens: topology analysis and user intent. The two techniques relying on topology (the two SchemeLens variants) were faster than the rectangular flat-top lens (with an improvement of 17% on average) and participants perceived this difference. P2.1, P2.2, P2.7 and P2.10 said they preferred the techniques relying on topology because there was no visible separation between magnified and unmagnified elements. However, surprisingly, taking into account user intent did not improve performance for any technique.

Memorization. A key aspect of the second study was to consider not only time performance but also the ability of users to memorize the diagram. Helping users to preserve a mental map of the diagram when working on a complex system may reduce cognitive load and thus improve the performance of high-level tasks. Our results suggest that exploiting both topology and user intent (SchemeLens with UI) helps users to better memorize the diagram than when using the rectangular flat-top lens. This result may seem surprising because SchemeLens with UI was also faster than the rectangular flat-top lens: Although participants spent less time interacting with the diagrams, they also better memorized them when using SchemeLens with UI.

Readability. Participants rated all techniques similarly for the ease of seeing links between *close* components. This is not surprising as fisheye lenses provide magnification in the focus area. However, users found it easier to see links between *distant* components with

SchemeLens (median score of 4.0/5 vs 2.0/5). To increase the readability of the elements in the focus area, SchemeLens does not occlude elements nor introduce distortions but slightly moves elements. This increased readability may explain why this technique is faster and favors the memorization of the diagram. However, a possible drawback of this approach is that it may seem less intuitive and less predictable than a fisheye view to some users (P2.6 and P2.10 in our experiment). Moreover, our pilot studies suggest that this approach is valuable only if the topology of the diagram is maintained (orthogonality, symmetries, etc.).

Spatial and Temporal Stability. Because SchemeLens is vector-based rather than pixel-based, it was necessary to smoothly change the focus when interacting to avoid spatial and temporal discontinuities. No participant mentioned a lack of smoothness while interacting, which indicates that the technique provides sufficient continuity between the magnification and the topological lens.

One drawback of the current SchemeLens implementation is that small cursor movements can cause more significant changes to distant components. However, this was considered as disturbing for only 3 of 26 users. Moreover, according to subjective preferences, users did not seem to feel more sickness and fatigue than with other techniques. Our informal observations are that observers more frequently observe this effect. We suspect this discrepancy may be due to the piloting effect, as when drivers are typically less susceptible to motion sickness than passengers due to a tighter control loop (e.g., [24]).

Empty space. In our application context, empty space does not convey a specific meaning. In other contexts, however, such empty space may be meaningful, as is the case with metro maps [4]. While our results suggest that users had a good overview of the global structure of the diagram using SchemeLens, it could be interesting to test the technique on diagrams where the absence or presence of empty space may matter.

Scalability to Larger Networks. In our application context, we need to be able to handle diagrams that would be printed out on an A3 sheet of paper. For security reasons, we had access only to simplified schematics. As such, we ran our experiments in a small window on about a quarter of a 20-inch screen, thus simulating a higher data density. Moreover, many applications require screen space for additional controls or linked views. Visually, SchemeLens should scale up to larger diagrams for which the zoomed-out overview remains understandable. For larger diagrams, some combination of semantic zooming and/or aggregation (e.g., as suggested by [18]) or using different levels of detail as in [11] may help.

Influence of topological rules. We use a set of topological rules to try to maintain a readable diagram, such as by maintaining orthogonality, same-edge alignment, and redundant loop alignment. One special case arises with bundled edges, in which two different orthogonal edges might partially overlap. Without specific handling, when nodes shift positions, these edges may become unbundled, leading to some suboptimal edges routings. We handle this case by introducing virtual, invisible nodes at such intersections. This approach maintains edge bundling, but does not eliminate all surprising edge routings. Additional topological rules should help reduce or eliminate these cases.

7 CONCLUSION

We introduce SchemeLens, a content-aware, vector-based fisheye technique. It uses the topology of the network diagram to scale up components based on their geometric proximity to the cursor and their topological proximity. Moreover, it maintains a topologically equivalent drawing and enforce aesthetics rules by using a space-making strategy to make room for scaled components. Optionally, by considering the direction of the mouse motion, it can also adapt the shape of the focus area according to users intents.

We validated this approach experimentally by two user studies. They showed that SchemeLens is significantly faster than both round and rectangular flat-top fisheye lenses for path-following tasks and that the SchemeLens variant taking user intentions into account favors an implicit memorization of the diagram.

Comparing SchemeLens with other fisheye techniques would be interesting to extend the results we obtained in our evaluations by taking other features into account. Furthermore, in a follow-up version, it would be worth adding topological information about paths, such as the direction of an electrical or hydraulic flow. This would allow improving the representation when taking user intentions into account, for instance by highlighting or magnifying the nodes that are in the direction of the flow from the current position of the cursor.

REFERENCES

[1] J. Alvina, C. Appert, O. Chapuis, and E. Pietriga. Routelens: Easy route following for map applications. In *Proc. AVI '14*, pages 125–128. ACM, 2014.

[2] L. Bartram, A. Ho, J. Dill, and F. Henigman. The continuous zoom: a constrained fisheye technique for viewing and navigating large information spaces. In *Proc. UIST '95*, pages 207–215. ACM, 1995.

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

[4] J. Bottger, U. Brandes, O. Deussen, and H. Ziezold. Map warping for the annotation of metro maps. In *IEEE Pacific Visualization Symposium*, pages 199–206, 2008.

[5] J. Brosz, S. Carpendale, and M. A. Nacenta. The undistort lens. In *Proc. EuroVis '11*, pages 881–890. Eurographics Association, 2011.

[6] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Making distortions comprehensible. In *Proc. VL '97*, pages 36–45. IEEE, 1997.

[7] S. Carpendale, J. Ligh, and E. Pattison. Achieving higher magnification in context. In *Proc. UIST '04*, pages 71–80. ACM, 2004.

[8] P. A. Fishwick. *Aesthetic computing*. MIT Press, 2008.

[9] G. W. Furnas. Generalized fisheye views. *SIGCHI Bulletin*, 17(4):16–23, 1986.

[10] E. R. Gansner, Y. Koren, and S. C. North. Topological fisheye views for visualizing large graphs. *IEEE TVCG*, 11(4):457–468, 2005.

[11] Hauser. Generalizing focus+context visualization. In *The Visual Extraction of Knowledge from Data*, pages 305–327. 2006.

[12] W. Huang, P. Eades, and S.-H. Hong. Layout effects: Comparison of sociogram drawing conventions. Technical Report 575, School of Information Technologies, University of Sydney, 2005.

[13] S. Kim, I. Woo, R. Maciejewski, D. S. Ebert, T. D. Ropp, and K. Thomas. Evaluating the effectiveness of visualization techniques for schematic diagrams in maintenance tasks. In *Proc. APGV '10*, pages 33–40. ACM, 2010.

[14] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM ToCHI*, 1(2):126–160, 1994.

[15] K. Marriott, H. Purchase, M. Wybrow, and C. Goncu. Memorability of visual features in network diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2477–2485, 2012.

[16] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

[17] T. Moscovich, F. Chevalier, N. Henry, E. Pietriga, and J.-D. Fekete. Topology-aware navigation in large networks. In *Proc. CHI '09*, pages 2319–2328. ACM, 2009.

[18] B. Musial and T. Jacobs. Application of focus + context to uml. In *Proceedings of the Asia-Pacific Symposium on Information Visualisation - Volume 24*, APVis '03, pages 75–80, 2003.

[19] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. Jellylens: Content-aware adaptive lenses. In *Proc. UIST '12*, pages 261–270. ACM, 2012.

[20] C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, 1995.

[21] H. C. Purchase, D. Carrington, and J.-A. Allder. Empirical evaluation of aesthetics-based graph layout. *Empirical Software Engineering*, 7(3):233–255, 2002.

[22] U. Rauschenbach, S. Jeschke, and H. Schumann. General rectangular fisheye views for 2d graphics. *Computers & Graphics*, 25(4):609–617, 2001.

[23] G. G. Robertson and J. D. Mackinlay. The document lens. In *Proc. UIST '93*, pages 101–108. ACM, 1993.

[24] A. Rolnick and R. E. Lubow. Why is the driver rarely motion sick? the role of controllability in motion sickness. *Ergonomics*, 34(7):867–879, 1991.

[25] M. Sarkar and M. H. Brown. Graphical fisheye views. *Commun. ACM*, 37(12):73–83, 1994.

[26] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: A metaphor for viewing large layouts on small screens. In *Proc. UIST '93*, pages 81–91. ACM, 1993.

[27] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM ToCHI*, 3(2):162–188, 1996.

[28] R. Spence and M. Apperley. Data base navigation: an office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54, 1982.

[29] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A Survey on Interactive Lenses in Visualization. In *EuroVis State-of-the-Art Reports*, pages 43–62. Eurographics Association, 2014.

[30] F. Wang, Y. Li, D. Sakamoto, and T. Igarashi. Hierarchical route maps for efficient navigation. In *Proc. IUI '14*, pages 169–178. ACM, 2014.

[31] N. Wong, S. Carpendale, and S. Greenberg. Edgelens: An interactive method for managing edge congestion in graphs. In *Proc. InfoVis '03*, pages 51–58. IEEE, 2003.

[32] I. Woo, S. Kim, R. Maciejewski, D. S. Ebert, T. D. Ropp, and K. Thomas. Sdviz: A context-preserving interactive visualization system for technical diagrams. In *Proc. EuroVis '09*, pages 943–950. Eurographics Association, 2009.

[33] A. Zanella, M. S. T. Carpendale, and M. Rounding. On the effects of viewing cues in comprehending distortions. In *Proc. NordiCHI '02*, pages 119–128. ACM, 2002.