Information Visualization
XX(X):1–13
©The Author(s) 2021
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Topology-Aware Space Distortion for Structured Visualization Spaces

Weihang Wang,[1] Sriram Karthik Badam,[2] and Niklas Elmqvist[3]

Abstract

We propose *topology-aware space distortion* (TASD), a family of interactive layout techniques for non-linearly distorting geometric space based on user attention and on the structure of the visual representation. TASD seamlessly adapts the visual substrate of any visualization to give more screen real estate to important regions of the representation at the expense of less important regions. In this paper, we present a concrete TASD technique that we call ZoomHalo for interactively distorting a two-dimensional space based on a degree-of-interest (DOI) function defined for the space. Using this DOI function, ZoomHalo derives several areas of interest, computes the available space around each area in relation to other areas and the current viewport extents, and then dynamically expands (or shrinks) each area given user input. We use our prototype to evaluate the technique in two user studies, as well as showcase examples of TASD for node-link diagrams, word clouds, and geographical maps.

Keywords

Topology-aware navigation, zooming, distortion, web-based visualization, user study.

Introduction

Geometric space provides the fabric, essence, and backdrop of a data visualization. However, not all space is created equal. The salience of a visualization varies across the visual substrate based on a range of factors, including color, shape, pattern, texture, and even the current interest of the user. As a result, all visual representations will have areas of greater and of lesser importance. Even though maintaining consistent information density is a common design principle,[1] the dynamic nature of human attention means that some graphical decisions cannot be made at design time because their governing factors vary based on the user's visual exploration. For example, while a node-link diagram is often laid out to minimize crossings and maintain uniform link distances,[2] the user may develop an interest in a subset of the graph when exploring the data, rendering all other nodes unimportant. Traditional navigation such as zooming and panning are often impractical in supporting changes in attention due to easily causing a loss of overview.[3]

What if we could use our structural knowledge of the visual representation and the user's interest to intelligently distort the visual space to give more screen real estate to important regions at the expense of unimportant regions? Inspired by topology-aware navigation,[4] which utilizes structural information to optimize navigation, we propose *topology-aware space distortion* as using topological knowledge of the visual representation to distort geometric space in support of the user's attention. In terms of the node-link diagram example discussed above, this would involve isolating the locations for the cliques of interest in the graph and then maximizing the space allocated to these nodes while shrinking the layout allocations of remaining nodes. Similar distortion mechanisms can be applied to visual space in other visualizations including word clouds, scatterplots, and geographical maps.

To exemplify our new concept, we present ZoomHalo, a technique for topology-aware distortion of two-dimensional visual spaces. The technique is defined by a three-step pipeline: (1) identify areas of interest in a visual representation; (2) calculate the amount of free visual space around each area of interest that does not encroach on other areas; and (3) expand each area of interest depending on the user's interest. Areas of interest are derived based on a *degree-of-interest* (DOI) function defined for the visual representation and the user's current attention; examples of areas of interest include the mouse cursor, the location of the user's gaze, or salient components of a visual representation, such as cliques in a node-link diagram or clusters in a scatterplot. We then tesselate the visual space using a Voronoi diagram based on these areas and use the computed cells to determine how far each area of interest vertex can be expanded.

Common zooming and distortion techniques used for both traditional and semantic zooming[5] include magnifying lens (fisheye hemi-spherical views)[6], fisheye lens[7] and hyperbolic visualization[8]. In contrast, ZoomHalo keeps a distorted overview of the visual representation while expanding multiple regions of interest, thus enabling seeing an overview and comparing the details within different parts

[1] University at Buffalo, Buffalo, NY, USA; [2] Apple Inc., Cupertino, CA, USA; [3] University of Maryland, College Park, MD, USA

Corresponding author:

Niklas Elmqvist, College of Information Studies, University of Maryland, College Park, MD 20742, USA.

Email: elm@umd.edu

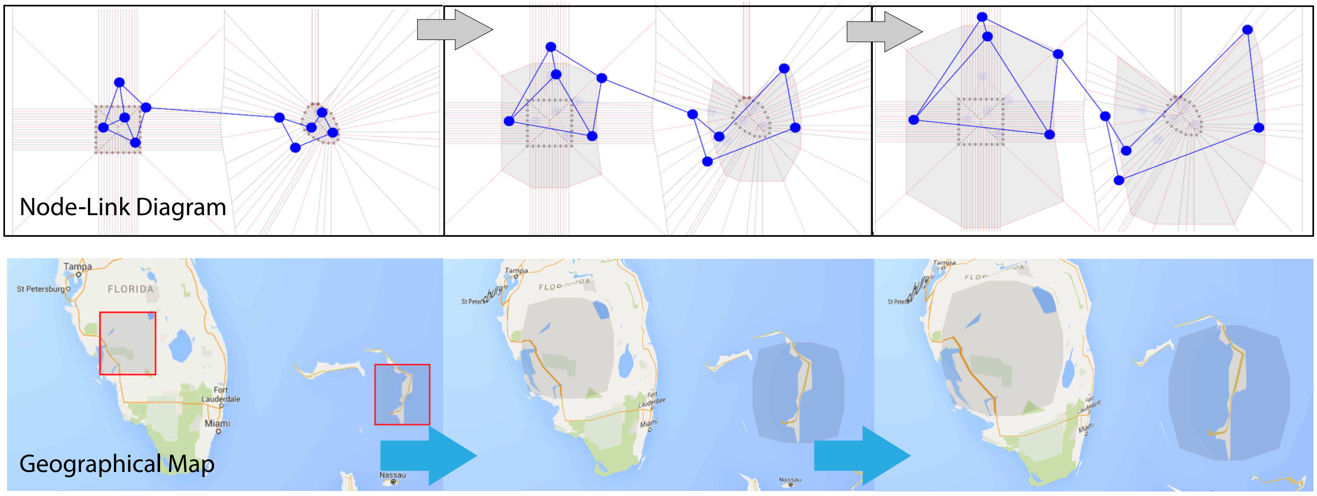


Figure 1. ZoomHalo is a topology-aware distortion technique to generate more space for multiple areas of interest in a visualization. (Top) An illustration of ZoomHalo applied to graph visualization. Two interest regions are selected (rectangle and freeform), Voronoi cells are calculated from them, and the nodes in the interest regions are spread out on zooming. The enlarged regions are shown in the background with a gray filling. (Bottom) A practical example of ZoomHalo with Google Maps.

of a visual representation simultaneously. For example, while searching multiple locations on a Google Map using zoom-and-pan interaction, we often lose the context of each search as soon as we move to a different search location. With topology-aware distortion, the locations can be treated as areas of interest that enlarge/zoom, while global attributes such as distance and position of the search locations can be still retained in the remaining distorted map to a perceivable degree.

To validate the technique, we have built a JavaScript library called ZOOMHALOJS that provides topology-aware space distortion for any visual substrate. We used the ZoomHaloJS implementation to apply to several visualizations and also in a quantitative user study involving a multifocus and multiscale visual search task on geographical maps. Through this study, we compared ZoomHalo with a baseline zoom-and-pan interaction provided by Google Maps in terms of search time and ability to retain the position and orientation of the search targets. Our results show a significant improvement in completion time using ZoomHalo. As a followup, we also compared ZoomHalo with a standard magnification lens technique that applies a spherical distortion to magnify specific areas of interest without distorting the overview, through an online study with a larger population pool. ZoomHalo, which leads to distortion on zooming, performed comparably to the magnifying lens for the target finding and retention tasks.

Background

Visual space is an intrinsic component of visualization, and grows in complexity in sync with the scale and complexity of the data being visualized. Much work exists on navigating and interacting with visualizations, but none to our knowledge take advantage of the topology of the visual space to intelligently distort it in support of the user's attention. Below we survey relevant work within general navigation, off-screen awareness, space distortion, and topology-aware navigation.

General Navigation

Standard navigation techniques such as zooming and panning support the user's dynamically changing attention by yielding control of the position and size of the viewport.[3] However, general navigation is plagued by an intrinsic tradeoff[7]: zooming in to see details causes loss of overview, while zooming out causes loss of fine detail.

The straightforward solution to this problem is to create one or more additional viewports for overview and details.[9] Such techniques solve the immediate problem, even for multi-scale spaces,[10] and have been shown[11] to be more efficient than both pan and zoom as well as fisheye views (below) for some tasks. However, providing multiple viewports[12] requires the user to split their attention between them, consumes additional screen space, and the spatial relation between the viewports is often not clear.

An alternative solution to the loss of overview is to provide the user with an awareness of off-screen objects even when zoomed in for details. Halo,[13] Wedge,[14] and EdgeRadar[15] give visual indications of the presence and sometimes the distance to off-screen targets. However, position and surrounding context of each target are lost. To remedy this, techniques such as Vacuum,[16] WinHop,[17] and Dynamic Insets[18] show graphical proxies of off-screen targets with their neighborhood. Despite this, off-screen visualization techniques only provide a local rather than a global overview, and there is a limit to how many targets can be shown.

Beyond these general navigation techniques, such as panning, zooming, multiple viewpoints, and off-screen awareness, some techniques resort to distorting the visual space. We discuss these next.

Space Distortion

More sophisticated solutions are possible if we are willing to relax the linear topology of the space. More specifically, non-linearly distorting the visual space allows for integrating one or several focus regions within an overview of the space itself. Generalized fisheye views[7] are the canonical example of these so-called focus+context techniques; related works include Carpendale and Montagnese's elastic presentation framework,[19] Pietriga and Appert's Sigma lenses,[20] and Shoemaker and Gutwin's multiple fisheyes.[10] However, the drawback with these techniques is that they all distort the space, thus rendering the position visual variable[21] less effective (which is a problem for many visualization techniques), and often cause instability in both the visual output space as well as the motor input space.[22]

Instead of applying distortion using a lens, some techniques instead distort the visual space directly. The rubber sheet stretching metaphor[23] provides handles on the visual space itself that the user can drag to non-linearly adapt the graphical representation. Accordion Drawing[24], based on guaranteed visibility property introduced by TreeJuxtaposer[25] where certain landmarks can be specified to always be visible, extends the rubber sheet metaphor to a general framework for 2D spaces. Mélange[26,27] is another approach to guarantee visibility at specific zoom levels by directly folding intervening space into 3D. Finally, hyperbolic trees[28] lays out hierarchies on a hyperbolic plane and renders it on a circular display; Munzner later applied this idea to large graphs in 3D space.[29] However, just as above, these techniques are highly visually unstable and relax the linear mapping of the substrate.

More recently, JellyLens[30] used focus+context lenses that adapt to the shape of the object of interest and magnify them by distorting a transition region between the focus regions (interest areas) and the context (the rest of the graphic). While this technique preserves the context to some extent, it can cause discontinuities between focus and context areas when multiple interest areas (spread out across the visual representation) are present, due to the transition regions. Furthermore, JellyLens requires geometric information of the underlying areas for adapting to the interest areas.

These distortion techniques are all highly relevant to our proposed topology-aware space distortion method and the ZoomHalo technique presented in this paper. However, instead of using local lenses, such as fisheye views or JellyLenses, or global mappings, such as hyperbolic trees, ZoomHalo uses a hybrid approach that distorts the global space based on one or more local focus region. Compared to JellyLens, which also uses specialized focus shapes, the ZoomHalo technique will use the entire global space to optimize visibility of the foci.

Topology-Aware Navigation

A recent family of interaction techniques called *topology-aware navigation* take advantage of the structure of the visual space to improve navigation; the intuition is that if we know something about the visual representation that the user is navigating in, we can make their navigation more efficient. This concept was first introduced by Moscovich et al.,[4] and exemplified for node-link visualization of graphs in the link-sliding technique, where the user's navigation is restricted to the linear paths connecting the nodes, and in the bring-and-go technique, where the immediate neighborhood of a particular node can be temporarily brought within close proximity to ease navigating to a neighboring node. Ghani et al.[18] combine the idea with off-screen navigation to create dynamically updating insets showing the surrounding context for destinations of links leaving the edge of the screen. Finally, recent work has begun to apply these ideas to graph visualization in particular. iSphere[31] maps a large graphs to the surface of a Riemann sphere that better preserves the graph topology and context. Similarly, structure-aware fisheyes[32] optimizes graph edges to minimize distortion for when viewing very large graphs.

The structure of the visual space can also be utilized to optimize the user's input in motor space (i.e., the space of the input device being used for navigation).[22] Semantic pointing[33] manages the ratio between control and display based on the distance to potential targets, essentially shrinking empty areas and growing targets in input if not visual space. Elmqvist and Fekete[34] extend this same idea to pointing in 3D space. Similarly, gravity navigation[35] uses an attention-gravity model to aid multiscale navigation by attracting (and repelling) zoom and pan operations based on the underlying visual structure. Wang and Chi[36] provide a topology-aware solution to dynamically generate metro maps by enlarging the best route to a destination and simplifying the rest of the map. The rest of the map locations are transformed to a simplified layout.

The above approaches to topology-aware navigation serve as the main source of inspiration for the topology-aware space distortion (TASD) proposed in this paper. However, unlike previous approaches, our new method uses topology awareness to non-linearly distort the visual space for optimizing navigation. We describe how below.

Topology-Aware Space Distortion

We define *topology-aware space distortion* (TASD) as the use of structural knowledge about a visual space to allocate more screen real estate to important areas at the expense of less important areas by dynamically distorting the space. This concept consists of two specific mechanisms that a concrete TASD technique must realize:

1. **Identification:** Where are the *areas of interest* in the visualization (and, by elimination, the areas of lesser interest)?

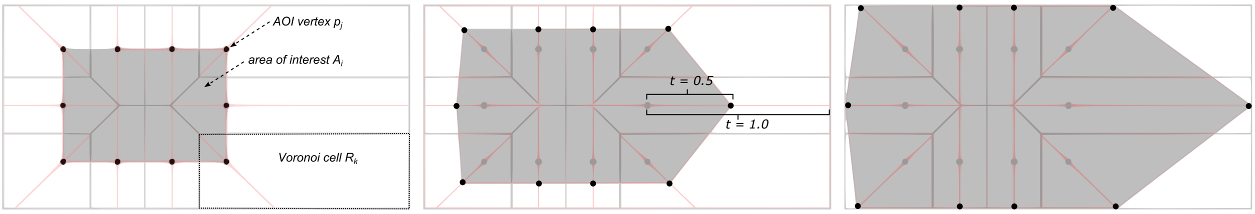
2. **Distortion:** How should visual space be distorted to allocate more screen real estate to areas of interest?

Identification

Identifying areas of interest is the phase where both the dynamic nature of the user's attention as well as structural knowledge of the visual representation can be taken into account. A general model for identifying areas of interest is by using a dynamic degree-of-interest (DOI) function defined for every point in a visual space. Given such a function, we can for example use a marching squares algorithm to determine areas of interest as 2D polygons on the space for a specific interest threshold.

Distortion

By definition, any region of the visual space that is not part of the identified areas of interest is of lesser importance. The purpose of the distortion phase is to non-linearly alter the visual representation to allocate more screen space to areas of interest by decreasing the space allocated to other parts. The



(a) No distortion; $t = 0$. **(b)** Some distortion; $t = 0.5$. **(c)** Full distortion; $t = 1.0$.

Figure 2. The ZoomHalo technique for a simple rectangular area of interest from no to full distortion. Note that we approximate the free space using a traditional Voronoi diagram that uses sampled vertices of the area of interest as generating sites. Each vertex is the expanded along the computed vertex normal (red line) until it reaches the border of the Voronoi cell.

key advantage of TASD is that this is done without changing the viewport, thereby retaining both the overview and context of the visualization.

Naturally, non-linear distortion may lead to instability in both visual and motor space. Certainly, the fact that topology-aware space distortion changes the structure of the visual substrate in response to data topology and user interaction directly affects any spatial mappings on this substrate. This means that TASD should not be used for visual representations that rely heavily on positional mappings to convey meaning.[21] Furthermore, another potential drawback with TASD's visual instability is that it prevents the user from building a mental map of the space. One solution to address both problems is for a TASD implementation to provide a mechanism for the viewer to understand the space distortion and its impact on the visual representation, i.e., a *distortion visualization*.

The ZoomHalo Technique

The ZoomHalo interaction technique is a concrete distortion technique for topology-aware space distortion that, given specific areas of interest represented as closed 2D polygons, calculates the free space around each area, and then dynamically expands these areas within their free space. In other words, the technique encompasses only the second phase—distortion. The name stems from the fact that the technique calculates the available "halo" of free space around each area of interest while taking other areas, their halos, and the extents of the space into account.

Below we describe each step of the technique, including deriving areas of interest, calculating the free space, and distorting the visual representation (Fig. 2 and Fig. 3). We also present some implementation notes for our JavaScript prototype of ZoomHalo.

Areas of Interest

The ZoomHalo technique does not specify an interest identification step; instead, any suitable approach can be used, such as a DOI function with marching cubes, user-identified areas, or representation-driven areas. In the end, the technique expects as input a set of areas of interest $I = \{A_1, A_2, \ldots, A_n\}$, where each area $A_i \subseteq W$ for the 2D visual space W, and $n \geq 1$. The most useful representation of an area A_i is as a closed polygon list of 2D vertices with clockwise winding: $A_i = \{p_1, p_2, \ldots, p_m | p_j \in \mathbb{R}^2\}$.

Calculating Free Space

For the sake of simplicity, we define the notion of *free space* in a 2D visual space $W \subseteq \mathbb{R}^2$ given a set of interest areas I as $W \setminus I$; i.e, any space that is not part of an area of interest is free space. However, in reality, this may depend on the type of content within the visual representation. For maps, selected regions can be areas of interest while the remaining can be simply treated as free space, and, on the other hand, if there was a pre-defined measure of the emptyness of a space (e.g., oceanic regions are free space) then the amount of free space changes. More interesting, however, is the amount of free space F_i associated with each area of interest A_i; intuitively, the amount each area of interest can be expanded without encroaching on the free space of other areas. If we assume that each area of interest has equal weight, it is reasonable to define the free space F_i for an area of interest A_i as all points in W that are closer to A_i than any other area. More formally, given a function $d(p, A)$ for calculating the Euclidean distance between a point $p \in \mathbb{R}^2$ and a set $A \subseteq \mathbb{R}^2$, we define free space F_i as follows:

$$F_i = \{p \in W \mid d(p, A_i) < d(p, A_k) \text{ for all } i \neq k\}.$$

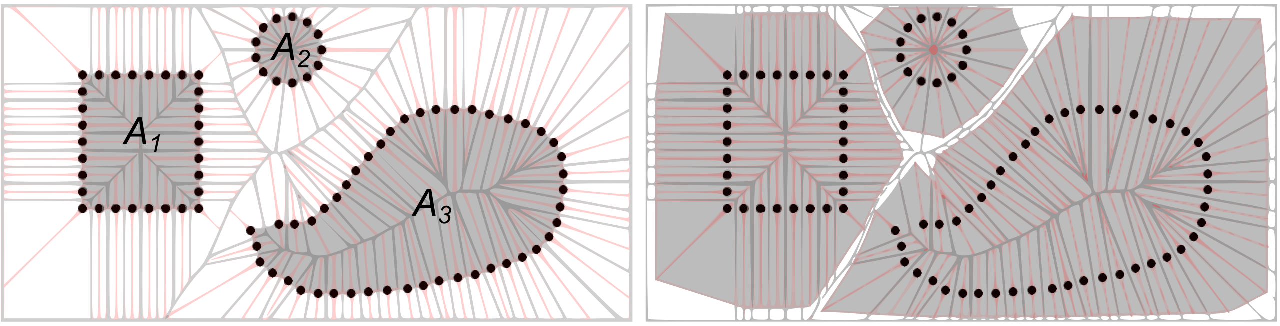
The most common way to calculate these "cells" of free space is using the Voronoi diagram, which calculates a spatial subdivision based on a set of generating sites. While standard Voronoi diagrams are defined for point sites, there exists generalizations in the literature to higher dimensions, weighted sites, as well as uncertainty[37]. Another approach would be to approximate the free space F_i for an area of interest A_i by summing standard Voronoi cells generated from point sites:

$$F_i = \bigcup_{k=1}^{j} R_k$$

where R_k is the Voronoi cell generated by site p_k in the point list for the polygon defined by the area of interest A_i.

Distorting into Free Space

The ZoomHalo technique grows each area of interest A_i into its corresponding free space F_i using a parameter $t \in [0, 1]$. This parameter gives the user interactive control over the amount of distortion; 0 means no distortion, whereas 1 means full distortion. In its simplest form, this could be achieved by linear interpolation, where the distorted area D_i for area of interest A_i is defined as $D_i = tA_i + (1 - t)F_i$. In practice,



(a) No distortion.

(b) High distortion.

Figure 3. Complex example of ZoomHalo being used in a visual space with three areas of interest.

however, such a formulation does not fully utilize the free space around each area of interest.

We therefore further constrain the problem by specifying that the boundary of the area of interest A_i should grow along its normal in order to maintain the area's general shape. Using the polygonal representation of A_i, this is achieved by computing the normal for each polygon vertex, and then computing the intersection between the normal and the boundaries of the free space cell F_i. This intersection consists of two points: the internal one is inside the area of interest and of no importance, but the external one expresses how far outwards that polygon vertex can be expanded along the normal before going outside the free space assigned to the vertex.

We can now use zoom parameter t to linearly interpolate between the original position of each vertex and the external intersection. This leads to the area grows from its original extents to (appx.) fit the shape of the free space. However, expanding each vertex in the area of interest polygon may lead to non-linear distortion (e.g., a rectangle non-linearly expands to a hexagonal shape in Fig. 2) since there is likely different amounts of free space around different parts of the area. This may be a problem when visualizing a texture, such as a geographical map, where the contents of each area of interest should remain undistorted. For this reason, we also implement a version with all vertices of each area expand the same amount. If enabled, this option means that the area of interest will grow uniformly and no distortion inside the area will occur, thus, making the area of interest retain its original shape with larger area.

Interaction Techniques

Several aspects of ZoomHalo can be controlled interactively, including the area of interests as well as the zoom parameter t. For the former, the user could for example directly use the mouse to define areas of interest, such as by drawing rectangles, circles, or general polygons on the visual space. Another approach may be to attach a nimbus around the mouse cursor that automatically makes its surroundings an area of interest. Finally, a gaze tracker would allow for directly letting the user's attention specify the area of interest. Beyond these interactions, the zoom parameter t could be indirectly controlled using a UI widget such as a slider. To mirror the notion of zooming without changing the viewport, another idea may be to associate t to the mouse wheel (or a pinch gesture).

Visualizing Distortion

One of the design tradeoffs on TASD was that distortion causes visual and motor instability and prevents the user from forming an accurate mental map of the space. To mitigate this, we propose to make the user aware of the shape and amount of distortion that has been applied to a particular visual space. Such *distortion visualizations* could use contour lines similar to those used in topographic maps, where contours inside an area of interest would be spaced out whereas those outside would be compressed to indicate the distortion in certain applications.

Implementation: ZoomHaloJS

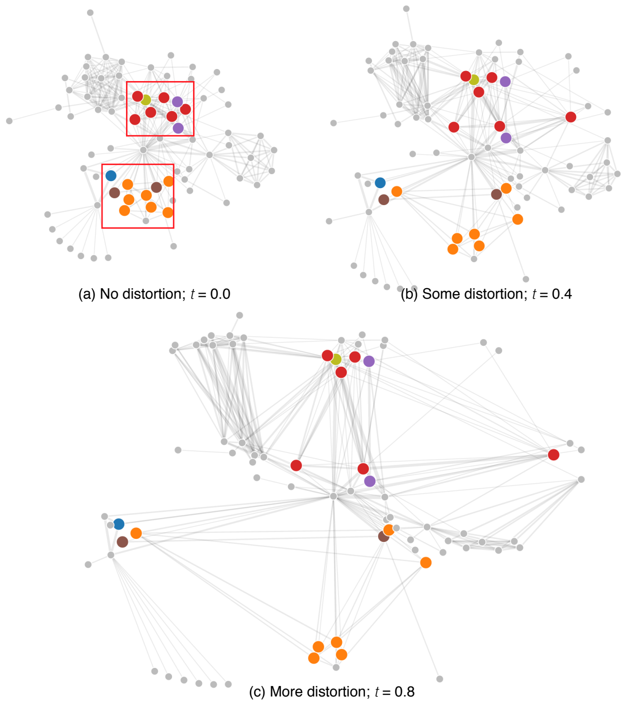
Our implementation of ZoomHalo is called ZoomHaloJS and is prototyped as a graphics-independent JavaScript library that can be combined with existing toolkits such as D3.[38] Our implementation uses the polygon-based representation to specify areas of interest. We also use the approximation discussed above where the free space for an entire area of interest is calculated as the union of the Voronoi cells for the vertices of the area of interest polygon. To ensure accurate approximation, we resample the polygon at regular intervals. We integrated the toolkit with examples built using WebGL, SVG, and HTML5 Canvas.

Examples

We demonstrate how ZoomHalo can be used in a wide range of applications by presenting examples we built using our ZoomHaloJS implementation: node-link diagrams, word clouds, treemaps, and maps.

Node-Link Diagram

Graph visualization using node-link diagrams are of particular interest for ZoomHalo because they often exhibit a globally sparse and locally dense topology that makes the ZoomHalo technique especially powerful. The topology of the graphs including the position of the nodes and the edges can be manipulated with ZoomHalo to spread out nodes in the areas of interest, while pushing the rest towards the visual boundaries of the visualization and between the areas of interest. Fig. 4 shows an example of how areas of interest



(a) No distortion; $t = 0.0$

(b) Some distortion; $t = 0.4$

(c) More distortion; $t = 0.8$

Figure 4. (a) A node-link with two interest regions zoomed (b and c) using ZoomHalo.

can be defined in the node-link diagram, enabling the empty space in the visual space be better utilized with ZoomHalo.

When the zoom parameter (t) is zero, the node-link diagram with its original layout is shown and the user can interact by selecting areas of interest. The interest regions (rectangles in Fig. 4) are then sampled into a polygonal shape to compute the Voronoi diagram from the points on the polygon. Following this, the normals are calculated for each Voronoi cell and used to expand the interest areas. When $t = 0.4$, the nodes in the interest regions start to spread out, and at $t = 1$ (maximum distortion), the nodes in the context regions are completely pushed to the visual boundary between regions of interest. For lower zoom levels (e.g., $t < 0.7$), the relationships of nodes in the context regions are mostly retained (however, non-linearly distorted). This example allows us look into complex subparts of the node-link diagram, while still understanding the features in the surrounding space.

Word Clouds

Our ZoomHalo-enabled Wordle[39] visualization (Fig. 5) enables viewing less frequency words hidden in the wordle layout without badly affecting the readability of other words. Also, it can dynamically provide more textual data detail as the user moves the cursor around the visualization. Similar to the previous example, the words surrounding the regions of interest are shrunk to expand the words within the regions. Since word clouds are packed representations, there are semantic and graphical zooming opportunities with ZoomHalo, based on the definition of empty space. For example, when the entire surrounding space including the words around the areas of interest are treated as areas of lesser interest (free space), this leads to a distortion of the words and the white space (Fig. 5). In contrast, a more

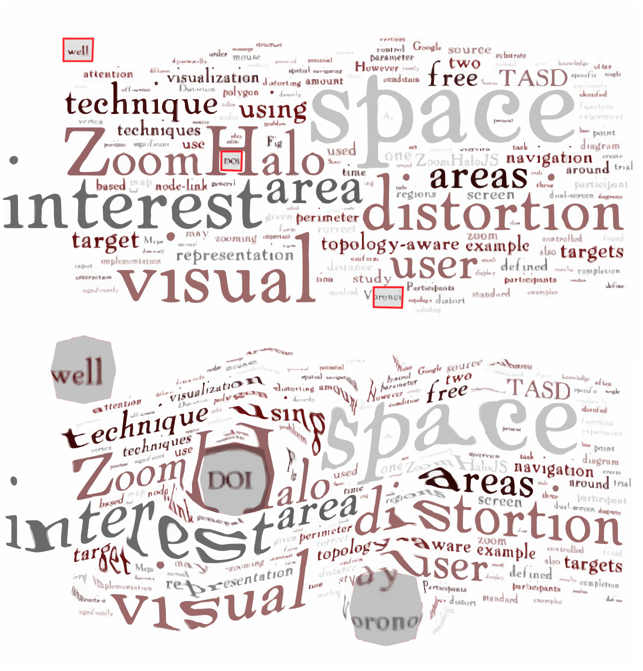


Figure 5. ZoomHalo-enabled word cloud navigation.

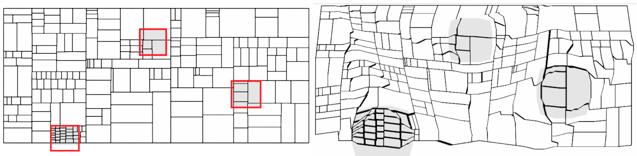


Figure 6. Treemap layout distorted by ZoomHalo to generate more space for regions of interest.

semantic approach for applying ZoomHalo would involve modifying the position and size of the words after distortion of only the white space using ZoomHalo. The latter approach is ideal for layout management, while the former retains the relative frequency between words to some extent.

Treemap

Treemaps[40] are popularly used for visualizing large hierarchial data, by allocating a fixed amount of space based on a parameter for each node and its children in a tree hierarchy. However, this also leads to regions in the treemap that are too dense and compactly packed to observe the substructures. ZoomHalo can modify the layout of a treemap to provide more space for regions of interest (e.g., dense areas in the treemap), by distorting the surrounding regions (Fig. 6). Figure 6 shows an example of the treemap layout distorted by ZoomHalo.

Map Visualization

Geographical maps often require visual comparison of multiple regions of interest. For example, to find a restuarant near a location, we often end up visually searching in multiple directions on a Google Map; however, switching from one direction to another leads to a loss of information regarding previous searches. While lens-based techniques and multiscale navigation with multiple overview+detail views are existing solutions, ZoomHalo can be applied to

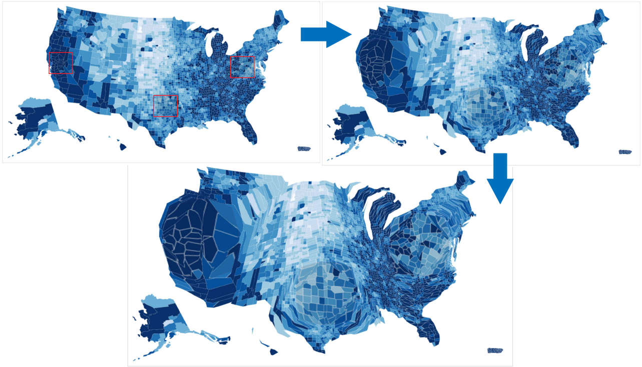


Figure 7. Zooming into multiple regions of interest in a choropleth map, using ZoomHalo, to compare them against each other. The high-level visual variables and spatial relationships such as distance, position, and spatial orientation between regions are retained to some extent (although distorted) without any discontinuities in the visual space.

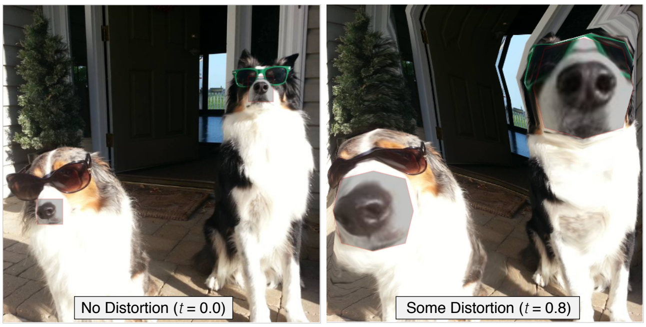


Figure 8. ZoomHalo applied to 2D raster images for zooming into two interest regions.

distort the entire context while expanding specific locations on the map. This yields a form of cartogram[41] representation of the original map.

For example, when applied to a choropleth map, ZoomHalo enables comparison of the distributions within a region of counties in California with other states (shown in Fig. 7), while retaining the high-level relationships such as the direction and distance between the interest regions.

Image-Space Distortion

The most straightforward use of ZoomHalo is to distort a 2D raster image based on user-defined interest regions. In contrast to visualizations, images lack free space and applying ZoomHalo can lead to a loss of meaning for the artifacts in the context regions (surrounding the interest areas). Fig. 8 shows an example of a photograph being distorted using ZoomHalo. This example was implemented using the ZoomHaloJS library and WebGL.

User Study: ZoomHalo vs. Dual-Screen

Our rationale for designing the TASD concept in general and the ZoomHalo technique in particular is that intelligently distorting a visual space may allow for simultaneously showing details about multiple areas of interest without losing the overall context. The intrinsic tradeoff is that the non-linear distortion may cause a loss of spatial orientation relations. To investigate this tradeoff between benefits and drawbacks, we performed a controlled user study involving this type of multifocus interaction[42] task, which involves two or more foci.

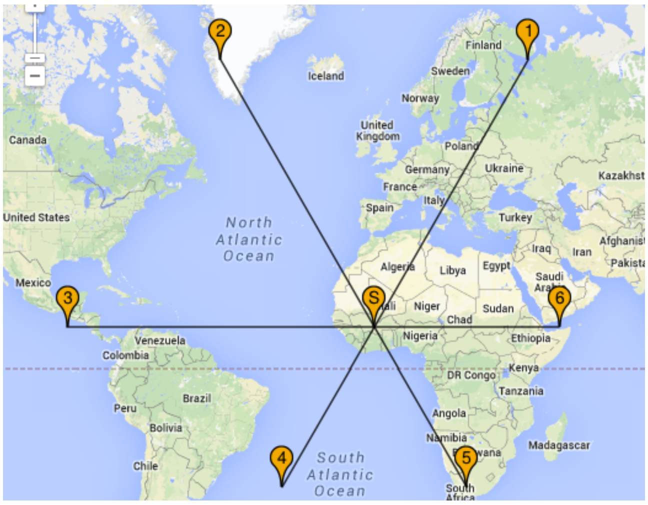


Figure 9. Basic study setup. The source target is marked S and is in the center of the star pattern. The labeling for the perimeter targets (1–6) was always consecutive but started at a random position, and their distance was randomized to be long or short. Figures 10 and 11 shows the two presentation conditions.

Our goal for this study was to compare ZoomHalo with panning and zooming in two different windows. It was particularly important to choose a non-distorted comparison because of our specific focus on investigating the impact of distortion for the ZoomHalo technique.

Participants

We recruited 16 (3 female, 13 male) participants from the undergraduate student population at our university to partake in the experiment. Participants were paid $10 upon completing a study session. Their ages ranged from 19 to 25 (median 23) years. All participants were self-reported proficient computer users (defined as using a computer on average at least 3 hours per day), had normal or corrected-to-normal vision, and were not color blind.

Apparatus

The experiment was performed on a standard desktop computer running the Microsoft Windows 7 operating system and equipped with a standard mouse (with wheel) and keyboard. The display was a 24-inch monitor set at 1920 × 1200 WUXGA resolution. The ZoomHaloJS software was running inside the Chrome web browser that was maximized on the screen. All user interaction was performed using the mouse.

Task and Dataset

The task involved a visual comparison between two or more focus regions, where participants were asked to find an identical target given a specific source target. More specifically, upon starting each trial, participants were confronted with 7 targets (Fig. 9): 1 source target in the center of the screen, and 6 targets arranged in a circle around the center and spaced at 60° intervals at two different distances from the center (the long distance twice the distance of the short one). The visual space itself was a multiscale Google Maps view (zoomed-out) that could be panned and zoomed. We picked Google Maps as this website is very familiar to most people and offer a baseline zoom-and-pan navigation; this was particularly important given our crowdsourced evaluation method. At the beginning of a trial, the view was zoomed out to show the entire map, and targets were represented as standard map markers.

The task was to find which of the six perimeter targets exactly matched the source target in the center. When zoomed in at least 10 magnification levels in Google Maps, each target (source and perimeter ones) changed from a single red-colored marker to a 3×3 matrix of circular balloons. The target changed back to a single marker if the map was zoomed out again. Each balloon was colored using one of nine colors (red, green, blue, cyan, orange, yellow, black, grey, magenta). Each perimeter target used the same number of colors as the source, with all of them except one (the correct one) having a single color permutation in the balloon matrix. This close resemblance made it virtually impossible to memorize the exact color configuration for each target, instead encouraging side-by-side comparison.

When the correct perimeter target had been identified, the participant selected the appropriate label corresponding to that target in a radio box and clicked a button to submit their answer. If their selection was incorrect, they were given visual feedback and asked to try again until the correct choice was given. We measured the time from the start of the trial until the correct answer was given, and encouraged participants to solve the trial as quickly as possible.

Immediately after finishing a timed trial (denoted by T1 for Task 1), the participant was given two additional follow-up tasks: (T2) recall the direction between the source and the correct perimeter target (represented as rotated arrows for each of the 60° increments used), as well as (T3) the distance between them (near or far). The purpose of tasks T2 and T3 was to determine the participant's spatial orientation in the visual space. To prevent participants using the target numbering to determine direction, perimeter targets were numbered consecutively but with a random starting point.

Experimental Factors

We used a factor M to represent the presentation method:

- DUAL-SCREEN: To model the standard way to solve this problem, this condition provided two independent Map views of the same trial, each exactly taking up half the available screen space (Fig. 10). Since the views were independent, users could control each of the views separately, often zooming in on the source and navigating the second between consecutive perimeter targets until the correct one was found. Figure 10 shows an example of the dual-screen condition.

- ZOOMHALO: The ZoomHalo condition used a single view consuming the entire available space where the user could left-click on the map to create a rectangular area of interest and rotate the mouse wheel to control the zoom parameter t (Fig. 11). Clicking on an area of interest a second time removed it. Standard zoom and pan were disabled for this condition. The participants were allowed to select one or more target regions around the source. The most common solution strategy here was to create an area of interest on the source target, change the zoom parameter to 1 (or close to 1), and then iteratively add and remove new areas of interest on perimeter targets until the correct matching one was found. Figure 11 shows an example of the ZoomHalo condition.

Note that the total available display space was kept constant for both techniques, but the different techniques used the space differently. For the Dual-Screen technique, the space was bisected into two equally-sized viewports. For the ZoomHalo technique, the single viewport used the entire available space.

Experimental Design

For search tasks such as this, random chance may have a major impact on completion time. To control this effect, we drew on a counterbalancing proposed by Pietriga et al.[43] to introduce a factor D that represented the discovery order that a participant found the correct target. With six potential targets around the source, D was defined to take all of the values 1, 2, 3, 4, 5, and 6. However, testing all the discovery orders along with repetitions can lead to a lot of tasks. Therefore, we chose a subset of these discovery orders for visual search tasks in each condition. Participants were told that the location of the correct target was random when it in fact was controlled by the value of D. This eliminated the risk of a participant being "lucky" (or "unlucky") all the time.

Time and error are perpetually opposed factors in visualization evaluation[44]; participants can either be tempted to speed through a trial to minimize their completion time but choose an incorrect answer, or check and double-check their answer to ensure correctness at the cost of speed. Requiring correct answers or capping the available time are two opposed approaches to eliminate one or the other. The idea of "luck control" is a method for controlling the impact of random chance when requiring correctness, and is a best practice pattern for visualization evaluation.[45] It is particularly important for visual search tasks such as this, where chance may otherwise have an overly significant influence.

With two experimental factors (M and D), we chose a within-subjects design and performed 2 repetitions with five visual search tasks on each condition, yielding a total of 20 trials per participant and a total of 320 recorded trials for the experiment. The order of presentation method was counterbalanced across participants to avoid systematic effects of practice whereas the discovery order was randomized. The discovery order subset, used for the five tasks in each condition, was balanced to make sure that we record readings for each of the discovery value. Dependent variables involved the completion time and the number of failures for T1 as well as correctness for T2 and T3.

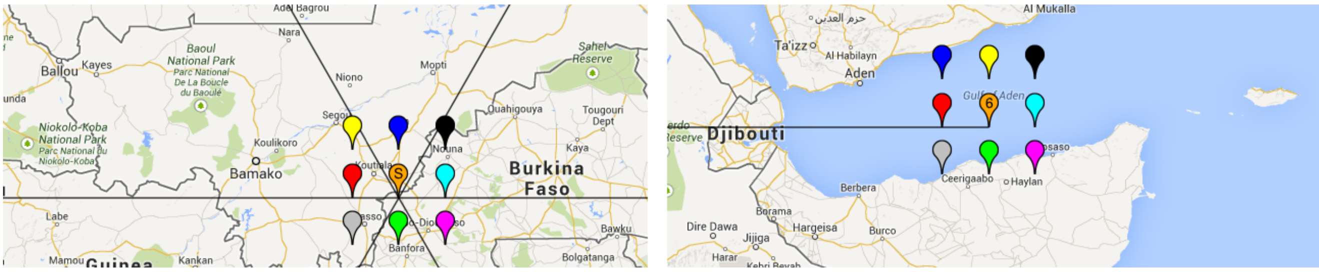


Figure 10. Basic study setup: dual-screen. Finding the correct target for the dual-screen condition. Each of the two side-by-side map views could be navigated independently. Figure 9 shows an overview of the task.

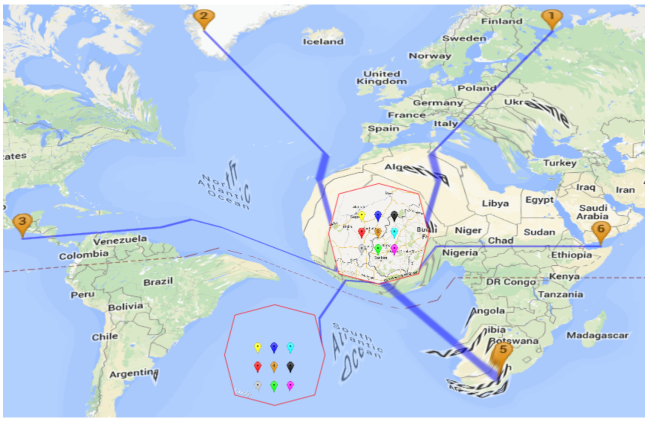


Figure 11. Basic study setup: ZoomHalo. Finding the correct target for the ZoomHalo condition. Note that we use uniform scaling of areas of interest created around perimeter targets to avoid distortion of targets. Figure 9 shows an overview of the task.

Procedure

An experimental session begun with a participant arriving in the laboratory and signing a consent form. The experiment administrator then explained the motivations for the work and presented the ZoomHalo technique. The task was reviewed and demonstrated for both the dual-screen and ZoomHalo techniques. This was followed by the two trial blocks, one per method. At the start of each block, the participant was allowed to train using the method three times before starting the timed trials. During training, they were allowed to ask questions and receive help. A trial both started and ended with a neutral screen during which the participant could rest. After finishing all trials for a block, the participant was asked to rate their experience on a subjective form with several Likert-scale questions. After finishing the entire trial, they were asked to give free-form written feedback. A single experimental session typically lasted 45 to 60 minutes.

Hypotheses

We formulate two hypotheses for the experiment:

H1 *Participants will be significantly faster with ZoomHalo than dual-screen for T1.* The fact that ZoomHalo eliminates the need to navigate the map display and enlarges multiple areas at once, will lead to significantly lower completion times.

H2 *Participants will have significantly worse performance with ZoomHalo than dual-screen for both T2 and T3.* The distortion caused by ZoomHalo will mean that participants will have a hard time recalling the target direction and distance.

Results

We analyzed our results using a multivariate analysis of variance for a repeated-measures experiment (i.e., relaxing the assumption for interdependence of observations; all other assumptions were valid). To increase statistical robustness, we eliminated all outlier measurements more than two standard deviations from the mean. The mean completion time for ZoomHalo was 37.13 (s.d. 10.58) seconds versus 42.89 (s.d. 9.91) seconds for dual-screen, a 13% improvement. This difference in presentation method M was significant; $F(1, 15) = 6.47, p < .012$. Not surprisingly, discovery order D also had a significant effect on completion time—$F(4, 15) = 28.34, p < .001$—with higher values yielding proportionally longer completion times.

We analyzed the two correctness measurements (angle versus distance) using logistic regression and found no significant effects of presentation method M; $F(1, 15) = .07, p = .79$ for angle and $F(1, 15) = .83, p = .58$ for distance.

Our results for the user study can be summarized as follows:

- Participants were significantly faster (13% on average) in completing the main search task with ZoomHalo than with dual-screen (confirming H1);
- Participants exhibited no significant difference in identification of both distance and position (direction) for the two conditions (rejecting H2).

Followup: ZoomHalo vs. Magnifying Lens

Our study showed that ZoomHalo outperformed dual-screen navigation in terms of time, but led to similar performance for recalling the distance and direction of the matching target. However, the presence of two views on the dual-screen is an aspect that can affect the participant performance for the target recall tasks. It was therefore unclear whether the similar performance is caused by presence of dual screens, or due to the distortion in ZoomHalo.

To further understand the distortion effects of ZoomHalo, we therefore conducted a follow-up online study with a larger population pool on Amazon Mechanical Turk. We had 80 participants (37 female, 43 male) for comparing

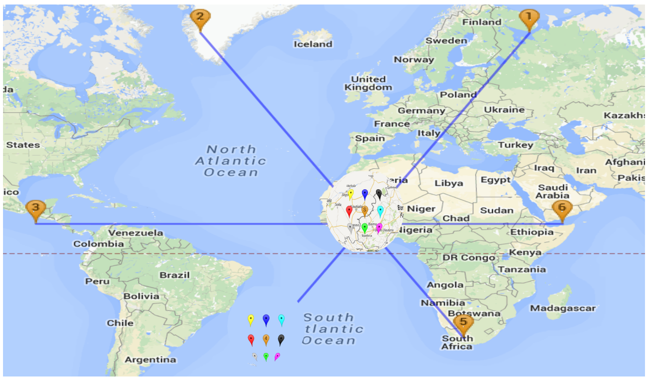


Figure 12. Basic study setup: magnifying lens. Finding the correct target for the magnifying lens condition. Figure 9 shows an overview of the task.

ZoomHalo with a circular magnification technique. The participants aged between 20 and 66, and they had (self-reported) experience interacting with a mouse or a touch-pad on a computer. All participants were Mechanical Turk users who have previously done at least 500 hits on Mechanical Turk with more than 97% approval rate. They were paid $0.9 for participation. This study had a between-subject design: 44 participants worked with magnifying lens condition and 36 worked with ZoomHalo (the difference was due to some participants not finishing the experiment).

Again, let us stress that we chose the undistorted magnifying lens precisely because it introduces no distortion, providing a strong contrast to the highly distorted ZoomHalo technique. Magnifying lenses are also a standard, well-established, and easily implemented navigation method employed in many interactive interfaces.[46]

The magnifying lens condition used a single view consuming the entire available space, just like the competing ZoomHalo technique. The participant selects a target to create areas of interest. A circular magnifying lens is then created on the target and the user can zoom using the mouse wheel or a pinch-to-zoom gesture. The magnifying lens creates a spherical distortion to fit the target into the circular region; however, this did not affect the readability of the pattern (9 circular balloons). In contrast to ZoomHalo, it keeps the context space undistorted and therefore the participants can better retain the spatial relationships for T2 and T3. We chose the same tasks as the previous study including (T1) comparing two or more regions to find the matching perimeter target given a specific source target and (T2, T3) recalling the distance and position of the correct target. The experiment design for each condition was also similar to the previous study. The completion time, failure count for T1, and the correctness for T2 and T3 were recorded for both conditions. An experiment session in this study lasted between 10 and 15 minutes.

The participants were allowed to compare one perimeter target at a time with the source while using the magnifying lens technique, since this technique by nature proposes zooming a single region of interest. On the other hand, participants in ZoomHalo could create multiple areas of interest on the perimeter targets and expand them all at once. Due to this difference, we expected the participants to have lower completion times with ZoomHalo. At the same time, we expected better performance for T2 and T3 in the magnifying lens condition, due to the distortion caused by ZoomHalo.

Contrary to our expectations, the difference between ZoomHalo and magnifying lens for completion time was not significant ($F(1, 79) = 1.51, p = .22$ from a multivariate ANOVA). The mean completion time for ZoomHalo was 76.5 sec. (s.d. 6.3) vs. 66.4 sec. (s.d. 5.34) for the magnifying lens. As expected, the discovery order (D) had a significant effect on completion time–$F(4, 79) = 24.70, p < .001$–with higher values yielding proportionally longer completion times.

A logistic regression on the two correctness measurements (T2, T3) showed no significant effects of method M; $F(1, 79) = .35, p = .557$ for angle and $F(1, 79) = .44, p = .509$ for distance. We found that the participants fail to identify the angle in 4.6% of the trials (on average) in case of ZoomHalo, while for magnifying lens it was 10.38%. On the other hand, the failure rate in recalling the distance was almost similar (ZoomHalo: 31.3%; Magnifying lens: 29.5%).

Discussion

Visual exploration is a process of interactively exploring different portions of the data embedded in visual representations.[47] Navigation, especially zooming and panning, is an important part of the exploration process. However, most navigation techniques do not take the topological information within the visual representation into account. This information could make the navigation process more efficient. While topology-aware navigation techniques do exist,[18,34,36] they are often limited to specific applications or visual structures.

Explaining the Results

ZoomHalo allows expansion of multiple areas of interest in the visual space simultaneously. For this reason, we expected ZoomHalo to be faster than dual-screen and magnifying lens. ZoomHalo indeed led to faster target search performance than dual-screen. This result is due to the repetitive process of zoom-and-pan interactions on Google Maps, leading to longer target matching time. On the other hand, when compared with a single magnifying lens condition, target search took a similar amount of time. We also noticed that the mean completion times in the onsite user study (ZoomHalo vs. dual-screen) were much lower than the completion time in the followup online study.

Upon further analysis, we realized that the computational and graphic capabilities of the computers used in the online studies were often lower than the onsite experiment apparatus. Therefore, the added delays in tessellation, as well as handling the continuous interaction of the user and rendering the transformed Voronoi cells, adversely affected the target search time in ZoomHalo vs. magnifying lens. Further studies are required to contemplate if ZoomHalo can lead to faster target search compared to other lens-based navigation techniques such as fisheye lens[7] and JellyLens.[30]

Global vs. Local Distortion

A contrast can be drawn between ZoomHalo and other magnification techniques based on how the areas of interest are magnified and how the rest of the space is shrunk. As identified in the related work, some of the popular magnification techniques include the Perspective Wall,[48] magnifying lens, generalized fisheye,[7] SigmaLens,[20] Mélange,[26] and JellyLens.[30] These magnification techniques can be distinguished on four aspects, inspired by the design goal distinction from Mélange[26]:

A1 *Focus visibility:* Multiple foci of any shape are visible simultaneously at the desired zoom level for any visualization.

A2 *Context visibility:* The context space is entirely visible.

A3 *Context awareness and visual continuity:* Foci are visually connected to context.

A4 *Distortion algorithm:* The distortion algorithm used on the foci and the context creates no information loss.

Table 1 summarizes how the techniques differ on these aspects.

Distortion algorithms have two forms of use: (1) for *global* layout manipulation where the focus (or foci) is magnified while compressing the entire context, and (2) for *local* magnification using focus+context lenses through a compressed transition region between the focus and the context. Generalized fisheye views are applied to an entire layout with a nonlinear distortion but they only support a single focus (using a distance and degree-of-interest definitions). On the other hand, focus+context fisheye lens can be applied to multiple foci but they create a transition region that affects the awareness of the immediate context surrounding the foci. JellyLens[30] creates a context-aware version of fisheye lens that adapts the lens shape and satisfies A1. Magnifying lenses lose a part of the context, thus, creating a visual discontinuity (A3). Overall, the transition region introduced by lens-based distortion techniques can make it harder to connect the focus and context. Beyond this, some local distortion techniques—SigmaLens—only support fixed shapes, thus, lacking in focus visibility aspects (A1).

Global distortion techniques—fisheye views, Perspective Wall, and Mélange—fall short in satisfying (1) focus visibility aspects by missing support for either multiple foci or flexible shapes, and (2) context visibility by distorting the context completely. ZoomHalo performs a global distortion, which is topology-aware. It ensures that multiple interest regions get more space by making use of less interesting regions in the entire visual representation. By doing so, ZoomHalo combines the best aspects of local and global distortion techniques to an extent by supporting multiple foci with arbitrary shapes and preserving the visual continuity (without a transition).

Applications and Limitations

Because of its hybrid local + global nature, ZoomHalo is better for visualizations that do not assign a meaning to position and have significant free space within the context. However, it is a double-edged sword for others: on one end, it can promise visual continuity in the entire space, but makes it harder to understand the graphical features within the distorted space. The continuity in the visual space, after distortion caused by ZoomHalo, helps retain some high-level spatial relationships such as distance and direction. More work is needed to understand which visual variables and spatial relationships can be retained/recalled after ZoomHalo's distortion.

When we compare ZoomHalo with fisheye and JellyLens techniques,[30] which create a transition region between focus and context areas, it is apparent that ZoomHalo's ideal application scenarios are different than others. JellyLens (and fisheye) is ideal when there is free space (e.g., oceans) closely surrounding the interest regions (e.g., and masses), as the free space will be part of the transition regions that are distorted. However, this can make it hard to understand relationships (e.g., distance) between multiple regions of interest due to the local distortion at the transition region.

Apart from ZoomHalo, multifocus interaction techniques[42] also tackle multiscale visual search. However, they provide completely different solutions involving multiple viewports and focus regions with representations of different scales. The drawback of having multiple viewports is the requirement of additional view space for each user interaction. In PolyZoom[42], this is observed in the form of adding new views to a hierarchical tree layout.

Finally, we would like to make a case for topology-aware space distortion techniques in general. Many techniques in the past have achieved magnification without full context distortion,[51] and some insist on avoiding distortion.[7] However, our results and examples showcase some benefits of ZoomHalo and prove that there is value to exploring topology-aware space distortion. We therefore think there is benefit to continuing this kind of work in the future.

Conclusion and Future Work

We have presented the ZoomHalo technique for topology-aware space distortion that intelligently distorts space given structural knowledge of the visual representation. We also described its three-step algorithm based on identifying areas of interest, calculating the free space, and distorting the visual representation. We have presented ZoomHalo examples for node-link diagrams, geographical maps, word clouds, and so on. A controlled laboratory experiment found significant improvement for the novel ZoomHalo technique compared to dual-screen configurations in a map-based visual search task. A followup study compared ZoomHalo with a magnifying lens to find that distortion has no effect on retaining distance and spatial orientation (direction).

We think there is much potential for additional exploration into topology-aware space distortion. In the future, we intend to also study this idea in conjunction with eye trackers, in 3D spaces, and for additional instantiations, visual representations, and datasets beyond the ZoomHalo technique.

Technique	References	A1	A2	A3	A4
Fisheye view	Furnas[49]	N (single focus)	P (distorted)	Y	P (nonlinear)
Fisheye lens	Furnas[49]	N (fixed shapes)	Y	P (transition)	P (nonlinear)
Perspective Wall	Mackinlay et al.[48]	N (single focus)	Y	Y	P (3D)
Magnifying Lens	Appert et al.[50]	Y	N	N (context lost)	–
SigmaLens	Pietriga & Appert[20]	Y	P (blended)	P (translucent)	–
Mélange	Elmqvist et al.[26,27]	N (fixed shapes)	P (distorted)	Y	P (3D)
JellyLens	Pindat et al.[30]	Y	Y	P (transition)	P (nonlinear)
ZoomHalo	N/A	Y	P (distorted)	Y	P (nonlinear)

Table 1. Effects of distortion techniques on focus and context (Y = Yes, P = Partially, N = No, – = not applicable). Aspect A4 is not applicable to magnifying lens and SigmaLens techniques because they do not distort but rather blend focus and context.

Acknowledgments

This work was partially supported by the U.S. National Science Foundation award IIS-1539534. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency. The authors also want to thank Corrie and Keegan, who gracefully agreed to sit for Figure 8.

References

1. Tufte ER. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press, 1983.
2. Purchase HC. Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing* 2002; 13(5): 501–516.
3. Furnas GW and Bederson BB. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 234–241.
4. Moscovich T, Chevalier F, Henry N et al. Topology-aware navigation in large networks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 2319–2328.
5. Perlin K and Fox D. Pad: An alternative approach to the computer interface. *Computer Graphics* 1993; 27: 57–64.
6. Leung YK and Apperley MD. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction* 1994; 1(2): 126–160.
7. Furnas GW. A fisheye follow-up: further reflections on focus + context. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 999–1008.
8. Lamping J, Rao R and Pirolli P. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 401–408.
9. Plaisant C, Carr D and Shneiderman B. Image browsers: Taxonomy and guidelines for developers. *IEEE Software* 1995; 12(2): 21–32.
10. Shoemaker G and Gutwin C. Supporting multi-point interaction in visual workspaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 999–1008.
11. Hornbæk K, Bederson BB and Plaisant C. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Transactions on Computer-Human Interaction* 2002; 9(4): 362–389.
12. Cockburn A, Karlson A and Bederson BB. A review of overview + detail, zooming, and focus + context interfaces. *ACM Computing Surveys* 2008; 41(1): 2.
13. Baudisch P and Rosenholtz R. Halo: a technique for visualizing off-screen objects. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 481–488.
14. Gustafson S, Baudisch P, Gutwin C et al. Wedge: clutter-free visualization of off-screen locations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 787–796.
15. Gustafson SG and Irani PP. Comparing visualizations for tracking off-screen moving targets. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. pp. 2399–2404.
16. Bezerianos A and Balakrishnan R. The vacuum: facilitating the manipulation of distant objects. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 361–370.
17. Partridge G, Nezhadasl M, Irani P et al. A comparison of navigation techniques across different types of off-screen navigation tasks. In *Proceedings of INTERACT*, *Lecture Notes in Computer Science*, volume 4663. Springer, pp. 716–721.
18. Ghani S, Riche NH and Elmqvist N. Dynamic insets for context-aware graph navigation. *Computer Graphics Forum* 2011; 30(3): 861–870.
19. Carpendale MST and Montagnese C. A framework for unifying presentation space. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. pp. 61–70.
20. Pietriga E and Appert C. Sigma Lenses: Focus-context transitions combining space, time and translucence. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 1343–1352.
21. Bertin J. *Semiology of Graphics*. University of Wisconsin Press, 1983.
22. Balakrishnan R. ”Beating” Fitts’ law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies* 2004; 61(6): 857–874.
23. Sarkar M, Snibbe SS, Tversky OJ et al. Stretching the rubber sheet: A metaphor for visualizing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. pp. 81–91.
24. Slack J, Hildebrand K and Munzner T. PRISAD: A partitioned rendering infrastructure for scalable accordion drawing (extended version). *Information Visualization* 2006; 5(2): 137–151.
25. Munzner T, Guimbretière F, Tasiran S et al. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. *ACM Transactions on Graphics* 2003; 22(3): 453–462.

26. Elmqvist N, Henry N, Riche Y et al. Mélange: Space folding for multi-focus interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 1333–1342.
27. Elmqvist N, Riche Y, Henry N et al. Mélange: Space folding for visual exploration. *IEEE Transactions on Visualization and Computer Graphics* 2010; 16(3): 468–483.
28. Lamping J and Rao R. The Hyperbolic Browser: A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing* 1996; 7(1): 33–35.
29. Munzner T. H3: laying out large directed graphs in 3D hyperbolic space. In *Proceedings of the IEEE Symposium on Information Visualization*. pp. 2–10.
30. Pindat C, Pietriga E, Chapuis O et al. JellyLens: content-aware adaptive lenses. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. pp. 261–270.
31. Du F, Cao N, Lin Y et al. iSphere: Focus+context sphere visualization for interactive large graph exploration. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, pp. 2916–2927. DOI:10.1145/3025453.3025628.
32. Wang Y, Wang Y, Zhang H et al. Structure-aware fisheye views for efficient large graph exploration. *IEEE Transactions on Visualization and Computer Graphics* 2019; 25(1): 566–575. DOI:10.1109/TVCG.2018.2864911.
33. Blanch R, Guiard Y and Beaudouin-Lafon M. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 519–526.
34. Elmqvist N and Fekete JD. Semantic pointing for object picking in complex 3D environments. In *Proceedings of Graphics Interface*. pp. 243–250.
35. Javed W, Ghani S and Elmqvist N. GravNav: Using a gravity model for multi-scale navigation. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*. pp. 217–224.
36. Wang YS and Chi MT. Focus + context metro maps. *IEEE Transactions on Visualization and Computer Graphics* 2011; 17(12): 2528–2535.
37. Sember J and Evans WS. Guaranteed Voronoi diagrams of uncertain sites. In *Proceedings of the Canadian Conference on Computational Geometry*. pp. 207–210.
38. Bostock M, Ogievetsky V and Heer J. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 2011; 17(6): 2301–2309.
39. Viégas FB, Wattenberg M and Feinberg J. Participatory visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics* 2009; 15(6): 1137–1144.
40. Shneiderman B and Wattenberg M. Ordered treemap layouts. In *Proceedings of the IEEE Symposium on Information Visualization*. p. 73.
41. Nusrat S and Kobourov SG. The state of the art in cartograms. *Computer Graphics Forum* 2016; 35(3): 619–642. DOI:10.1111/cgf.12932.
42. Javed W, Ghani S and Elmqvist N. PolyZoom: Multiscale and multifocus exploration in 2D visual spaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 287–296.
43. Pietriga E, Appert C and Beaudouin-Lafon M. Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 1215–1224.
44. Plaisant C. The challenge of information visualization evaluation. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, pp. 109–116. DOI:10.1145/989863.989880.
45. Elmqvist N and Yi JS. Patterns for visualization evaluation. *Information Visualization* 2015; 14(3): 250–269. DOI:10.1177/1473871613513228.
46. Hornbæk K and Frøkjær E. Reading of electronic documents: The usability of linear, fisheye, and overview+detail interfaces. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. pp. 293–300.
47. Tominski C. *Event-Based Visualization for User-Centered Visual Analysis*. PhD Thesis, University of Rostock, 2006.
48. Mackinlay JD, Robertson GG and Card SK. The Perspective Wall: Detail and context smoothly integrated. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 173–179.
49. Furnas GW. Generalized fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. pp. 16–23.
50. Appert C, Chapuis O and Pietriga E. High-precision magnification lenses. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, pp. 273–282. DOI:10.1145/1753326.1753366.
51. Carpendale S, Ligh J and Pattison E. Achieving higher magnification in context. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. pp. 71–80.