Structure-aware Fisheye Views for Efficient Large Graph Exploration

(Supplemental Material)

Yunhai Wang, Yanyan Wang, Haifeng Zhang, Yinqi Sun,
Chi-Wing Fu, Michael Sedlmair, Baoquan Chen and Oliver Deussen

Abstract—We provide the implementation details of the cluster lens and path lens in this supplemental material.

1 Cluster Lens & Path Lens

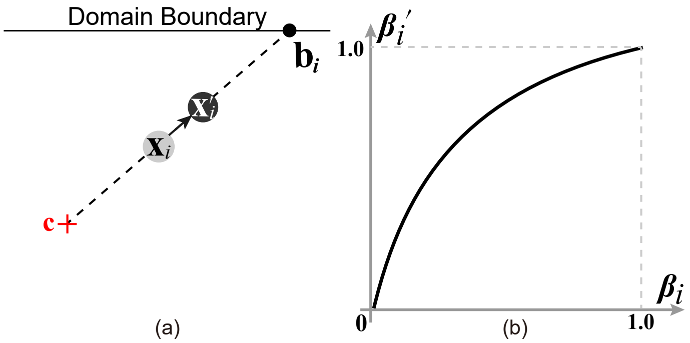

Fig. 1. Graphical fisheye: (a) each node $\mathbf{x}_i$ is moved away from the focal point $\mathbf{c}$ toward its boundary point $\mathbf{b}_i$ along a line that joins them; (b) distortion function ($m = 3$) for the distance ratio β_i; see Eq. (1) for details.

1.1 Background: Graphical Fisheye

Given a graph layout $\mathbf{X}$ and a focal point $\mathbf{c}$, a graphical fisheye magnifies the graph by displacing each $\mathbf{x}_i$ away from $\mathbf{c}$. The method first locates $\mathbf{b}_i$ on the domain boundary by extending a line from $\mathbf{c}$ through $\mathbf{x}_i$ (see Figure 1(a)) and then displaces $\mathbf{x}_i$ to $\mathbf{x}_i'$ along the line by the following equation:

$$\mathbf{x}_i' = \mathbf{c} + (\mathbf{b}_i - \mathbf{c})\beta_i', \text{ where } \beta_i' = \frac{(m+1)\beta_i}{m\beta_i + 1}\ ,\ \beta_i = \frac{||\mathbf{x}_i - \mathbf{c}||}{||\mathbf{b}_i - \mathbf{c}||}\ , \tag{1}$$

$m \geq 0$ is the parameter to control the magnification factor. Note that β_i is a distance ratio to be increased nonlinearly to β_i' (see Figure 1(b)); then, the method takes β_i' to create the new position $\mathbf{x}_i'$.

1.2 Cluster Lens

For exploring a cluster in a graph, the user often needs to examine not only the nodes and edges in the cluster but also how the cluster connects with other clusters. A desired lens should thus magnify the cluster of interest and provide the context of the entire graph. Previous fisheye views are based on one or multiple focal centers. Applying them, however, does not guarantee that all nodes of the cluster are properly magnified. Therefore, we developed the cluster lens, which allows users to specify a convex focal area and then magnify this region (and the cluster) linearly in the fisheye view, while compressing the outside context using the fisheye distortion model.

In our framework, we can easily support a cluster lens by modifying the way in which we produce $\mathbf{X}'$ from $\mathbf{X}$. Suppose $\hat{m} \geq 0$ the target magnification factor and $\mathbf{c}$ the centroid of the user-specified focal area, i.e., the focal center. For nodes inside the focal area, we compute $\mathbf{x}_i'$ as $(\hat{m}+1)\ \mathbf{x}_i$. For nodes outside the focal area, we compute $\mathbf{x}_i'$ through the graphical fisheye model, but inversely handle the magnification factor m

- *Y.H. Wang, Y.Y. Wang, H. Zhang and Y. Sun are with Shandong University. Email: {cloudseawang, yanyanwang93, hiphone.zhang,sunyinqi0508 } @gmail.com.*
- *B. Chen is with Peking University. E-mail: baoquan.chen@gmail.com.*
- *C.-W. Fu is with the Chinese University of Hong Kong. E-mail: cwfu@cse.cuhk.edu.hk.*
- *M. Sedlmair is with VISUS, University of Stuttgart, Germany. E-mail: michael.sedlmair@visus.uni-stuttgart.de.*
- *O. Deussen is with Konstanz University and Shenzhen VisuCA Key Lab, SIAT, China. E-mail: oliver.deussen@uni-konstanz.de.*
- *Y.H. Wang and Y.Y. Wang are joint first authors and H. Zhang and Y. Sun contribute equally to this work.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

in (Eq. 1), so that the magnification outside and inside the focal area seamlessly match each other. We first locate point $\mathbf{p}_i$ on the boundary of the focal area from $\mathbf{c}$ to $\mathbf{x}_i$ (see Figure 2(a)). By derivation we find that in Eq. (1) we are able to set

$$m = \frac{\hat{m}}{1-(\hat{m}+1)\cdot\gamma}, \qquad \text{with} \qquad \gamma = \frac{||\mathbf{p}_i - \mathbf{c}||}{||\mathbf{b}_i - \mathbf{c}||}.$$

Figure 2(b) shows a result of applying our cluster lens.

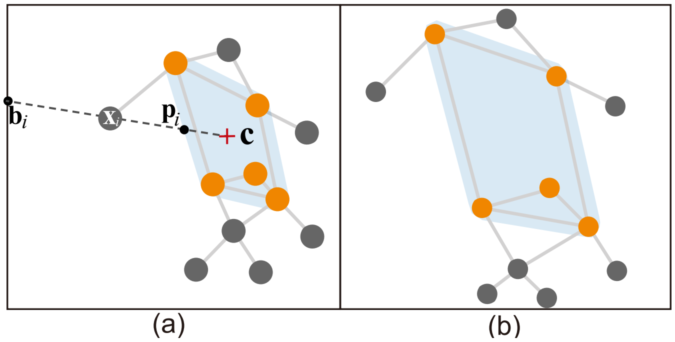

Fig. 2. Illustration of the cluster lens. (a) using $\mathbf{p}_i$ to inversely determine the magnification factor m in the graphical fisheye model (Eq. (1)). (b) Result generated by our cluster lens model.

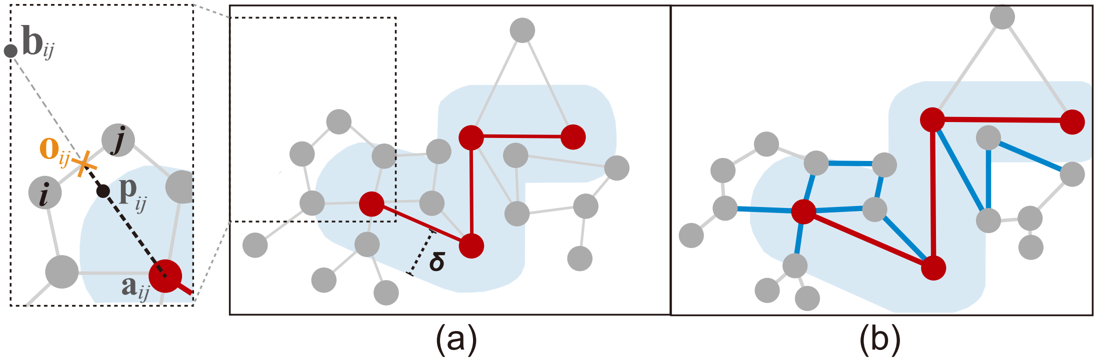

Fig. 3. Illustration of the path lens: (a) the path of interest (in red) and the focal region (in blue); (b) result produced by the path lens.

1.3 Path Lens

Path exploration tasks, such as exploring a path's neighboring nodes and checking their degrees, require a magnification subject to the path of interest. Yet, we are not aware of any fisheye technique that can support such nontrivial zooms. For this reason, we added a path lens to our framework for supporting such tasks.

To create a path lens, users can simply pick two nodes in the graph. Then, our method automatically finds and locates the shortest path that connects these nodes (red path in Figure 3(a)) and defines the focal area around the path (blue region in Figure 3(a)). Note that the focal area has a radius of σ measured from the path. We empirically set σ as $\sqrt{m}/28$ times the screen size, so that the focal area adaptively increases with the magnification factor m.

Given the focal area and $m \geq 0$, we define the edge constraints as follows to achieve a path constraint:

- We compute the midpoint of each edge, say $\mathbf{o}_{ij}$.
- For edges whose midpoints are inside the focal area, we modify their length d^s_{ij} as $(m+1)\cdot d^s_{ij}$ for achieving a linear zoom of the path.
- For edges whose midpoints are outside the focal area, we first find the shortest distance from $\mathbf{o}_{ij}$ to the path. Then we locate the closest point $\mathbf{a}_{ij}$ on the path, as well as point $\mathbf{b}_{ij}$ on the domain boundary along the line that joins the first two points; see the enlarged view in Figure 3(a). After that, we update the distortion factor d^s_{ij} by

$$d^s_{ij} = \frac{m+1}{m\beta_{ij}+1}\cdot d^s_{ij} \qquad \text{with} \qquad \beta_{ij} = \frac{||\mathbf{o}_{ij} - \mathbf{a}_{ij}||}{||\mathbf{b}_{ij} - \mathbf{a}_{ij}||}.$$