4-2020

Techniques to visualize occluded graph elements for 2.5D map editing

Kazuyuki FUJITA
Tohoku University

Daigo HAYASHI
Tohoku University

Kotaro HARA
Singapore Management University, kotarohara@smu.edu.sg

Kazuki TAKASHIMA
Tohoku University

Yoshifumi KITAMURA
Tohoku University

Techniques to Visualize Occluded Graph Elements for 2.5D Map Editing

Kazuyuki Fujita
Research Institute of Electrical Communication,
Tohoku University
Sendai, Japan
k-fujita@riec.tohoku.ac.jp

Daigo Hayashi
Research Institute of Electrical Communication,
Tohoku University
Sendai, Japan
dh9489@riec.tohoku.ac.jp

Kotaro Hara
School of Information Systems,
Singapore Management University
Singapore, Singapore
kotarohara@smu.edu.sg

Kazuki Takashima
Research Institute of Electrical Communication,
Tohoku University
Sendai, Japan
takashima@riec.tohoku.ac.jp

Yoshifumi Kitamura
Research Institute of Electrical Communication,
Tohoku University
Sendai, Japan
kitamura@riec.tohoku.ac.jp

CHI'20 Extended Abstracts, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6819-3/20/04.
http://dx.doi.org/10.1145/3334480.3382987

Abstract
We propose an interface with two novel techniques to visualize occluded graph nodes and edges that help the user edit map data with a 2.5D geographical structure (*e.g.,* multi-floor indoor maps). We first design a visualization technique —*Repel Signification*— that employs micro-animation to signify the graph elements that are overlapping with each other (and potentially erroneous). We also design a technique that enables the user to edit the occluded components with *Expansion Interaction*, which simultaneously visualizes both in-floor and across-floor occluded connections between the map elements. The combination of the two methods would enable the map editors (non-experts) to effectively find and fix erroneous data in 2.5D maps without changing the operation manner from the existing 2D map-editing interface.

Author Keywords
Geovisualization; Graph visualization.

CCS Concepts
•Human-centered computing → Interaction techniques; Geographic visualization;

Introduction
Location-based services utilize geographical data of the physical environment (*e.g.,* road networks, points-of-interest)

Figure 1: (a) Node/edge occlusions and across-floor connections make it difficult to find topological errors when a map for a multi-floor building was manually composed. (b) We design a technique that visualizes these occluded elements with micro-animation (*repel*) to support users to find them. We also design an interaction technique that visualizes (c) occlusions and (d) across-floor connection relationships related to the element near the cursor, which supports users acquire and edit the occluded elements.

to help users explore, navigate, and search for information about the physical world. Readily available map data (*e.g.,* OpenStreetMap[1]), represented as geographically and topologically constrained networks of nodes and edges (*i.e.,* graphs), enables the development of such technologies. On OpenStreetMap, for instance, the data contributors use a map editing tool to draw nodes and edges to extend the graph that represents the geography and topology of the built environment in the given region. However, existing editing tools have limited capabilities in helping users explore and edit 2.5D geographical data that emerge as a result of representing the indoor environment of multi-floor buildings and dense, complex urban structures where layers of geographical data are placed on top of each other. One consequence of such inept interfaces is our inability to fix erroneous map elements that are unobservable due to occlusions (*e.g.,* components that are incorrectly connected but hard to find; Figure 1a).

Considerable attention has been paid to addressing occlusions and consequent problems in network visualization, where researchers explored ways to visualize complex data like social networks, molecular maps, and transport networks [2, 12, 13]. For example, researchers have attempted to improve the visibility by rearranging the layout by using tree structures [21, 29], radial views [5], or fisheye distortions [24, 25], etc. However, much work has dealt with geographically unconstrained networks. A unique restriction of a map that requires nodes and edges to represent physical coordinates prevents us from optimizing the layout of graph elements (which mutates the elements' coordinates). Meanwhile, prior work has documented types of occlusions [4, 32] occurred in geographical data, and much also has been done on resolving the occlusions by bundling edges together [1, 14], applying a lens-metaphor visualization [33], or directly interacting with them [16, 26, 32]. However, little has been done on exploring how to deal with erroneous elements in 2.5D maps.

[1] https://www.openstreetmap.org/

In this paper, we propose novel methods to visualize occluded elements in the context of editing 2.5D map data. We design the methods with the intention to extend the existing 2D map interface. First, to support users find occluded elements, we design a visualization technique that uses micro-animation (*Repel Signification*) (Figure 1b) to signify the occluded elements. Second, to enable users to edit (acquire) map elements that are overlapping each other, we implement an *Expansion Interaction*—see Figure 1c and 1d—which visualizes the occlusions and across-floor connection relationships related to the element near the cursor. We believe the combination of the two methods will help non-expert users find and fix the map data errors in 2.5D maps with almost no change of operation manner from the existing 2D map interface.

Background and Related Work

Modern location-based services and geographic information systems (GIS) demand high-quality spatial data for them to operate [20]. Mobile navigation tools like Google Maps[2], for example, combines a large and fine grained street network data with a routing algorithm to navigate a driver from one point to another. For such technologies to operate properly, data quality is paramount, and thus GIS and cartography researchers have studied spatial data quality for decades [3, 15, 27]. Van Oort offered a list of types of spatial data quality through a comprehensive survey [28].

Involvement of increasing number of non-experts without knowledge of geographical data emphasizes quality control

[2] https://maps.google.com/

of spatial data [28]. Involvement of paid or volunteer online workers to map geographical data (sometimes called volunteered geographical information or VGI) is common nowadays [6, 7, 8, 9, 10, 17, 18, 19, 23]. Notably, OpenStreetMap aims to create a set of public map data with edits of millions of volunteer workers [6, 7, 8]. Haklay *et al.* reported that the data provided by online workers is fairly accurate [7]. Recently, there are also attempts to apply OpenStreetMap to indoor route planning with 3D representation (*e.g.*, [30]). Priedhorsky *et al.* explored a technique to communicate potentially erroneous data in 2D maps of bicycle lanes. [18]. But we lack simple techniques to assess the quality of spatial data, particularly the geographical data that contains across-floor information and resides in 2.5D space.

Debiasi *et al.* have schematized visual clutter problems, which can occur when the components and geographic surfaces overlap with each other [4]. Wong *et al.* have also listed possible readability issues in edge congested graphs [32]. Although these visibility issues in 2D graph data are well investigated and summarized in detail, there are still few studies that consider geographic data with 2.5D structure. When dealing with 2.5D maps that consider multiple floor levels, across-floor connections (*e.g.*, stairs or elevators) can be utilized, but they cannot be essentially represented in most existing visualization techniques.

Communicating and visualizing complex graph data has been a research topic of information visualization. There are number of techniques to deal with the visual clutter problems by interacting with the graph data. Wong *et al.* have proposed techniques called EdgeLens [33], in which graph edges curve away from the user's point of focus allowing them to disambiguate the visual clutters without losing the nodes' spatial relationships. EdgePlucking offers the user to move edges with plucking action to reduce congestion [32]. Schmidt *et al.* have designed a group of techniques to solve visual clutter problems utilizing multi-touch interaction [26]. Riche *et al.* tries to improve visibility by interactively controlling link curvature [22]. These techniques resolve graph occlusions in 2D maps well, but they are sometimes not effective for use in 2.5D map editing tasks. One reason is that they do not consider 2.5D map structure. Another reason is that the visualization in these techniques requires the user to perform additional operations which sometimes conflict with the operations used in the existing map editing tools.

Proposed Techniques

Design Considerations

Supposing the task of data contributors co-editing 2.5D maps with multi-floor structure data, we design an interface to allow them to easily find and edit occluded components. Here we describe design considerations for the interface as follows.

Quick signification of the occluded elements. When a data contributor explores a map containing some occlusions (that are potentially topological errors), the interface should represent them to quickly notice to the user. One of the representative approaches is to change the visual properties (*e.g.*, color, shape, size, etc.) of the overlapping objects, which utilizes a human visual system called preattentive processing [11]. Our interface adopts a repelling micro-animation because many of visual properties such as color and shape are already used for representing other purposes in most commercial maps.

Detailed representation of the topology on the occluded elements. Existing tools like OpenStreetMap already offer map editing functionalities, but no way has been offered to

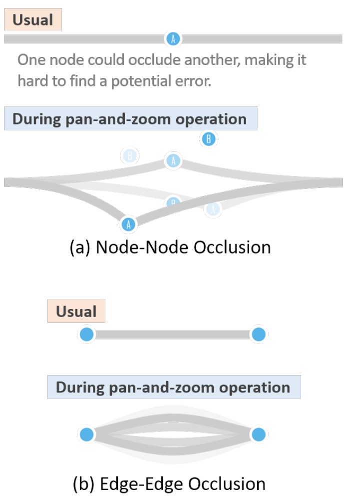

Figure 2: The behavior of *Repel Signification* for (a) node-node and (b) edge-edge occlusion. It uses shaking micro-animation to signify where an occlusion exists. Note: The translucent components represent animated time-lapse. The characters on the edges are not actually displayed.

edit and fix the overlapped components or across-floor connections. Since finding erroneous elements is relatively a difficult task that requires the user to understand the structural relationship between components [4], our interface makes the components visually understandable and clickable to help the user correct the topology.

Retention of spatial and topological relationships. Since geographical accuracy is essential in map data, so the interface should basically maintain the spatial and topological relationships between components. We minimize the effect of the micro-animation on the spatial/topological relationships by making it occur only during the user's pan-and-zoom operation, inspired by the interactive relief shearing technique proposed by Willett *et al.*[31].

Familiarity of the user interface. Editing map data requires more operation variations than just navigation, such as adding, moving, and deleting map elements, so no more operations should be added for visualizing occluded errors. Therefore, we apply a subtle add-on to familiar pan-and-zoom interface, which basically follows the style of existing map editing tools like OpenStreetMap.

Compatibility with PC. Of course map navigation and editing are used on various types of devices. As an initial exploration, this study focuses on designing mouse-based user interface on a PC.

Based on these considerations, we design methods to visualize and edit occluded map data on top of a familiar node-edge graph visualization, which resembles existing geographical data editing tools. First, to help the user find occluded data, we introduce *Repel Signification*, which signifies these elements with micro-animation coupled with panning and zooming operation (Figure 1b). Second, to enable editing the occluded components, our *Expansion Interaction* visualizes both in-floor and across-floor connections between the map elements near the cursor (Figure 1c). We implement these methods and the graph editing user interfaces using HTML, CSS, and JavaScript (D3.js). In this section, we suppose an specific environment to demonstrate our proof-of-concept prototype: interacting with a graph that has 3840 $\times$ 2400 *px* size and is generated on a grid with interval of 160 *px* though a 1280 $\times$ 800 *px*-size window on a PC.

Technique 1: Repel Signication
To notice the existence of occluded elements, this technique utilizes micro-animation to signify them; the occluded components are signified with an animation only when the user is performing pan-and-zoom operations (Figure 2). The basic idea of the algorithm is that the pairs of node-node, node-edge, and edge-edge that have close proximity are repelled and shaken using a metaphor of elastic objects like a spring. The repel force exponentially increases based on the proximity between the pair of elements, but these components remain swinging around the original position because the moving direction of these components are randomly determined at every frame.

In our implementation, the overlapping pairs are identified (and their repel forces are also calculated) by calculating each distance between the pair components (*i.e.*, node-node, edge-edge, or node-edge). We set the average distance of the repelled node and edge from the original position to approximately 16 *px* and 25 *px* respectively when the pair components completely overlap (and repel force is maximum). The shaking micro-animation gradually stops once the user stops dragging or zooming. Figure 2 shows example appearances of the repel motion.

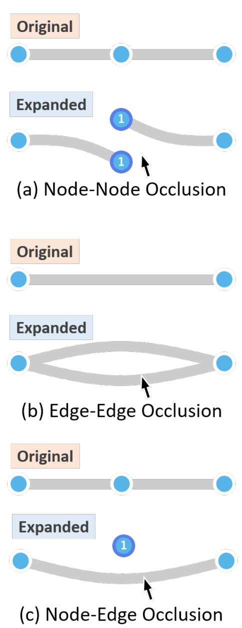

Figure 3: Behavior patterns of *Expansion Interaction* for in-floor overlaps.

Technique 2: Expansion Interaction
When a data contributor finds an erroneous element using *Repel signification*, then it is expected for him/her to fix it to the correct state by deleting/moving the element or adding new component(s). To achieve this, we introduce *Expansion Interaction*, a technique to visualize a specific element in 2.5D map, which tries to address problems of both difficulties in interpretation and acquisition of the occluded elements. With this technique, as the user moves the cursor around a graph, the component closest to the cursor is highlighted and identified as a *target component* for the expansion. Then the system visualizes its *related components* (*i.e.* components that are directly connected to the target component and their overlapped components) by showing them around the target component with repel animation while resolving their overlaps. Both the target component and the visualized related components become clickable, indicating that they can be edited by existing operations like ordinary map editing tools. We individually design the expansion behavior between in-floor and across-floor graph connections, as follows.

In-floor Expansion
Figure 3 overviews the basic behavior patterns of *expansion* interaction for in-floor overlaps which consist of node-node, edge-edge, and node-edge expansions. The component that is closest to the cursor (and closer than a certain threshold distance) is determined the target component. Accordingly, the system identifies its related components as follows: (i) the edges that are directly connected to the target component, and (ii) the overlapping components with (i) and the target component. These related components are visualized with repel animation so that the overlaps are resolved; each pair of overlapping components move apart each other, and the displacements are determined by the close proximity between them before expansion; in our implementation, the displacement gets exponentially larger based on the proximity and we adjusted the maximum displacement (when two components overlap completely) as approximately 20 *px*. Once the *expansion* occurs, the expanded components remain visible unless the cursor moves away from either of them at the threshold distance.

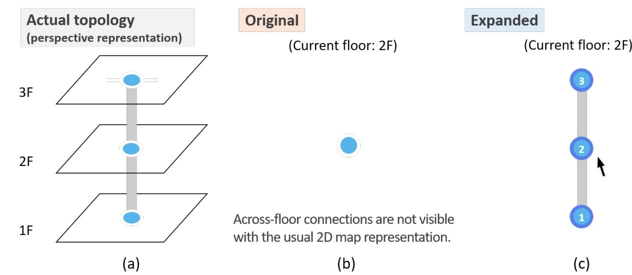

Figure 4: Behavior of *Expansion Interaction* for across-floor connections.

Across-floor Expansion
Figure 4 shows the basic behavior of expansion interaction for across-floor connection. To represent the across-floor connection relationships in 2.5D maps, we introduce to simply illustrate them two-dimensionally while maintaining floor-level relationships; upper-floor and lower-floor components are respectively drawn above and below the target component in a 2D map. In addition, a floor number is drawn on each of related nodes; we preliminary tested other representation patterns of the floor level such as change of the nodes' shapes or colors, but the method of describing floor numbers seemed to be the most easiest to understand the topology. In the example where the user is inspecting an error at an elevator node in the second floor as shown in Figure 4, only one node (second floor) is originally shown (Figure 4b), while the two other nodes with one level above and one level down become respectively shown at a distance

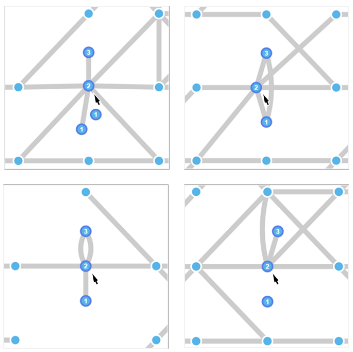

Figure 5: Example representations of simultaneous expansion for both in-floor and across-floor connections.

(40 *px* in our implementation) above and below the target component when the cursor is close (Figure 4c). Note that this expansion motion is not accompanied with repel animation in order to differentiate the visual effect between in-floor and across-floor representation.

In case there are both related components for in-floor and across-floor expansions in the graph, we simultaneously visualize them. When the user specifies a target component, the related components are determined as follows: (i) the edges that are directly connected to the target component in the floor, (ii) the nodes that are directly connected with the target component across different floors, (iii) the overlapping components with (i), (ii), and the target component, and (iv) the components that are prospectively overlapped with the newly displayed components of (ii) and (iii). The system then executes the expansion for all of these related components while resolving the overlaps and visualizing across-floor connection relationships at the same time. Figure 5 and video figure show several examples of simultaneous expansion for both in-floor and across-floor connections.

Early Results and Future Work

We conducted a preliminary test to get user feedback of the proposed techniques. Nine participants (6 males and 3 females) ranging from 20 to 27 ages participated. We first gave them an explanation of the purpose of the two techniques and how to use them, and then asked them to try the techniques in the context of correcting erroneous data in 2.5D maps. From the interviews after the test, we got many positive comments on the general user experience such as *easy to understand the topology on the occluded graph data*. The animation during *Repel Signification* was generally accepted, as some participants commented that it was *cute*, *friendly*, or *pleasant*, while one commented that *it might be annoying that the animation always occurs*. *Expansion Interaction* was also basically liked, as commented *node selection was easy and smooth*. Most of negative comments were about difficulties in selecting a edge error or its low visibility, even with the expansion. Some participants complained that the errors were hard to see when zoomed out.

In this study we presented novel techniques to visualize and edit occluded elements specified for 2.5D map data editing tasks. We designed *Repel Signification*, which signify and visualize the occluded components to support the user find them, and then we also designed *Expansion Interaction* to visualizes in-floor and across-floor connection relationships or overlaps relating to the occluded elements. Our future work includes investigating the fundamental performance through a target acquisition task, assessing the usability in more realistic tasks of correcting erroneous data in 2.5D indoor maps, and applying the techniques to touch-based devices.

Acknowledgements

This work was partly supported by the Cooperative Research Project Program of the Research Institute for Electrical Communication, Tohoku University.

REFERENCES

[1] Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, and Tim Dwyer. 2016. Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 541–550.

[2] Skye Bender-deMoll and Daniel A McFarland. 2006. The art and science of dynamic network visualization. *Journal of Social Structure* 7, 2 (2006), 1–38.

[3] Nicholas R Chrisman. 1991. The error component in spatial data. *Geographical information systems* 1, 12 (1991), 165–174.

[4] Alberto Debiasi, Bruno Simoes, and Raffaele De Amicis. 2016. Schematization of Clutter Reduction Techniques in Geographic Node-link Diagrams using Task-based Criteria.. In *VISIGRAPP (2: IVAPP)*. 109–116.

[5] Peter Eades. 1991. *Drawing free trees*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited.

[6] Mordechai Haklay. 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning B: Planning and design* 37, 4 (2010), 682–703.

[7] Mordechai Haklay, Sofia Basiouka, Vyron Antoniou, and Aamer Ather. 2010. How many volunteers does it take to map an area well? The validity of Linus' law to volunteered geographic information. *The cartographic journal* 47, 4 (2010), 315–322.

[8] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing* 7, 4 (2008), 12–18.

[9] Kotaro Hara, Shiri Azenkot, Megan Campbell, Cynthia L Bennett, Vicki Le, Sean Pannella, Robert Moore, Kelly Minckler, Rochelle H Ng, and Jon E Froehlich. 2015. Improving public transit accessibility for blind riders by crowdsourcing bus stop landmark locations with google street view: An extended analysis. *ACM Transactions on Accessible Computing (TACCESS)* 6, 2 (2015), 5.

[10] Kotaro Hara, Vicki Le, and Jon Froehlich. 2013. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 631–640.

[11] Christopher Healey and James Enns. 2011. Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics* 18, 7 (2011), 1170–1188.

[12] Jeffrey Heer and Danah Boyd. 2005. Vizster: Visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. IEEE, 32–39.

[13] Michael Kaufmann and Dorothea Wagner. 2003. *Drawing graphs: methods and models*. Vol. 2025. Springer.

[14] A. Lambert, R. Bourqui, and D. Auber. 2010. 3D Edge Bundling for Geographical Data Visualization. In *2010 14th International Conference Information Visualisation*. 329–335. DOI: http://dx.doi.org/10.1109/IV.2010.53

[15] Alan M MacEachren, Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elisabeth Hetzler. 2005. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science* 32, 3 (2005), 139–160.

[16] Tomer Moscovich, Fanny Chevalier, Nathalie Henry, Emmanuel Pietriga, and Jean-Daniel Fekete. 2009. Topology-aware Navigation in Large Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 2319–2328. DOI: http://dx.doi.org/10.1145/1518701.1519056

[17] Reid Priedhorsky. 2010. The value of geographic Wikis. (2010).

[18] Reid Priedhorsky, Mikhil Masli, and Loren Terveen. 2010. Eliciting and Focusing Geographic Volunteer Work. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10)*. ACM, New York, NY, USA, 61–70. DOI: http://dx.doi.org/10.1145/1718918.1718931

[19] Reid Priedhorsky, David Pitchford, Shilad Sen, and Loren Terveen. 2012. Recommending routes in the context of bicycling: algorithms, evaluation, and the value of personalization. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 979–988.

[20] Jonathan Raper, Georg Gartner, Hassan Karimi, and Chris Rizos. 2007. Applications of location–based services: a selected review. *Journal of Location Based Services* 1, 2 (2007), 89–111.

[21] Edward M. Reingold and John S. Tilford. 1981. Tidier drawings of trees. *IEEE Transactions on software Engineering* SE-7, 2 (1981), 223–228.

[22] Nathalie Henry Riche, Tim Dwyer, Bongshin Lee, and Sheelagh Carpendale. 2012. Exploring the Design Space of Interactive Link Curvature in Network Diagrams. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 506–513. DOI: http://dx.doi.org/10.1145/2254556.2254652

[23] Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, and Jon Froehlich. 2019. Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data At Scale. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 62, 14 pages. DOI: http://dx.doi.org/10.1145/3290605.3300292

[24] Manojit Sarkar and Marc H. Brown. 1992. Graphical Fisheye Views of Graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. ACM, New York, NY, USA, 83–91. DOI: http://dx.doi.org/10.1145/142750.142763

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

[26] Sebastian Schmidt, Miguel A. Nacenta, Raimund Dachselt, and Sheelagh Carpendale. 2010. A Set of Multi-touch Graph Interaction Techniques. In *ACM International Conference on Interactive Tabletops and Surfaces (ITS '10)*. ACM, New York, NY, USA, 113–116. DOI: http://dx.doi.org/10.1145/1936652.1936673

[27] David Sinton. 1978. The inherent structure of information as a constraint to analysis: Mapped thematic data as a case study. *Harvard papers on geographic information systems* (1978).

[28] Pepijn AJ van Oort. 2006. *Spatial data quality: from description to application*. Wageningen Universiteit.

[29] John Q Walker. 1990. A node-positioning algorithm for general trees. *Software: Practice and Experience* 20, 7 (1990), 685–705.

[30] Zhiyong Wang and Lei Niu. 2018. A data model for using OpenStreetMap to integrate indoor and outdoor route planning. *Sensors* 18, 7 (2018), 2100.

[31] Wesley Willett, Bernhard Jenny, Tobias Isenberg, and Pierre Dragicevic. 2015. Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3563–3572. DOI: http://dx.doi.org/10.1145/2702123.2702172

[32] Nelson Wong and Sheelagh Carpendale. 2007. Supporting interactive graph exploration using edge plucking. In *Visualization and Data Analysis 2007*, Vol. 6495. International Society for Optics and Photonics, 649508.

[33] Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. 2003. Edgelens: An interactive method for managing edge congestion in graphs. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*. IEEE, 51–58.