Semantic Resizing of Charts Through Generalization: A Case Study with Line Charts

Vidya Setlur*
Tableau Research

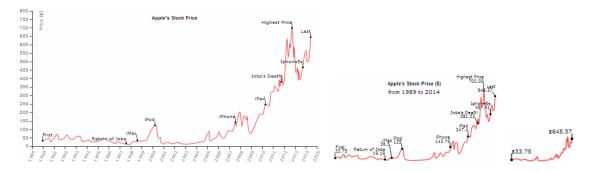Haeyong Chung†
University of Alabama in Huntsville

Figure 1: An example of a line chart automatically generalized to different display sizes. The algorithm preserves the various elements of the line chart based on their semantic importances at a given display size.

ABSTRACT

Inspired by cartographic generalization principles, we present a generalization technique for rendering line charts at different sizes, preserving the important semantics of the data at that display size. The algorithm automatically determines the generalization operators to be applied at that size based on spatial density, distance, and the semantic importance of the various visualization elements in the line chart. A qualitative evaluation of the prototype that implemented the algorithm indicates that the generalized line charts preserved the general data shape, while minimizing visual clutter. We identify future opportunities where generalization can be extended and applied to other chart types and visual analysis authoring tools.

Keywords: Responsive charts, visual clutter, spatial metrics.

Index Terms: Human-centered computing—Visualization—

1 INTRODUCTION

In the context of presenting data visualizations, resizing is particularly critical when engaging with a dashboard featuring limited screen real-estate, and/or when visualizations created on one display must then rendered on a different-sized display. The major challenge associated with developing techniques that facilitate resizing and creating multi-scale visualizations, is the significant number of variations that must be considered to represent a rescaled visualization effectively. A visualization author may need to represent every detail of the visualization at a given display size, while also taking into account every possible combination of the display properties such as resolution, size, and aspect ratio.

General resizing techniques such as uniform scaling and scale-and-stretch can be easily applied to resizing a visualization, but they tend to make the visualization illegible and increase the amount of visual clutter at smaller scales [46]. Such resizing also does not consider semantic information represented by the underlying data. Hence, it is crucial for visualization techniques to support a smarter

---

*e-mail: vsetlur@tableau.com
†e-mail: haeyong.chung@uah.edu

way to automatically adapt visual representations so that the viewer can read the chart more easily, regardless of the particular display size.

In cartographic literature, *generalization* refers to the process of abstracting the visual detail in a map, with the goal of maintaining the legibility of the map at any given scale [35]. Inspired by cartographic generalization, this work explores how these principles can be applied to resizing data visualizations effectively at different scales. Particularly, we extend cartographic generalization to rescaling line charts, a common chart type. We present a set of spatial metrics to examine geometric properties and relationships among elements in a line chart. These metrics are used to determine the presence of visual clutter and complexity in a view. Based on these metrics, we developed a set of generalization techniques for semantically resizing line charts to target display sizes.

2 RELATED WORK

There are several approaches to resizing content beyond mere uniform scaling and can be organized into the following categories:

2.1 Cartographic Generalization

Cartographic generalization enables one to simplify or eliminate less semantically important features, exaggerate more important ones, and resolve visual clutter to improve information quality on a smaller scale [11, 30, 35]. Route maps have effectively and succinctly applied various forms of generalization [3, 47]. Kray et al. describe a method of presenting route instructions on a mobile device depending on various situational factors such as limited resources and varying quality of positional information [28]. Although automatic map generalization techniques have been described for two decades in the cartographic and geographic literature, little research has been undertaken on how these methods could be extended to other forms of visualization.

2.2 Scroll, Pan and Zoom, Overview + Detail

Scroll and pan behavior maintains the layout and size of the original viewport, allowing for a portion of the visualization to be seen at any given time [2, 1], resulting in a loss of context [6]. Semantic zoom lets the user see different amounts of detail based on semantic importance [16]. Panning and zooming are often combined to create a continuous navigation experience [7]. Overview + detail tech-

niques display more than one level of detail such that manipulating the overview causes a corresponding change in the detail view. In SeeSoft [19], a miniaturized overview of text operates as a scrollbar for a more detailed view. While overview + detail reduce disorientation by having the overview co-present with the detail view, the techniques often suffer from visual discontinuity between the two views. Schwab et al. [34] studied how pan and zoom timelines are used to navigate large time series data. They found that the visual context and orientation play a role in efficient navigation. While these interaction techniques provide users the flexibility of viewing a portion of the content at a time, the navigation can be often disorienting especially when the frame of reference is not clearly defined. Our work explores how line charts can be generalized for a specific target size, displaying detail for more salient information.

2.3 Content Retargeting

For viewing large-sized content on smaller screens, there has been work on adapting or *retargeting* the content for these smaller displays. Automatic reformatting of web pages concatenates columns or hides less important information [15, 39]. With the prevalance of online data journalism, Kim et al. [25] explored how data thumbnails in these articles can be designed to be interpretable. MiniMap [33] changes the size of the text relative to the size of the viewport. Other techniques explore changing the geometrical properties of visualization elements and views, including geometrical transformation [4, 41, 43] and view deformation [46, 5]. Vistribute [22] automatically distributes visualizations and user interface components across multiple heterogeneous devices. Shi et al. [36] evaluate a diversity of automatic line simplification algorithms based on positional accuracy and processing time. These various techniques for retargeting however, may get challenging if the layout transformations affect the semantics of the content. We explore generalization as a way to preserve the recognizability of semantically important elements in the line chart at different display sizes.

2.4 Responsive Visualizations

The prevalence of mobile devices motivates the need to design communicative visualizations that are responsive to varying screen sizes. Kim et al. [24] identify strategies and trade-offs between the presentation of information and the intended takeaway of the chart. Often these visual representations and graphical user interfaces have to be adapted for smaller displays and lower resolution [8, 18, 12]. ViSizer [46] presents a perception-based framework for scaling important regions uniformly and deforming homogeneous context. MobileVisFixer [45] adapts a reinforcement learning-based approach that automatically learns and applies decision rules for generating mobile-friendly visualizations. Hoffswell et al. [21] analyzed a corpus of responsive news visualizations that informed a prototype tool for designing responsive visualizations for different device contexts. Our work is motivated by similar goals, but we specifically explore how cartographic generalization principles can be applied to line charts. By considering the spatial relationships of the individual elements in the chart, we developed an automated approach to emphasize and de-emphasize information based on their semantics and target display size.

3 ALGORITHM

Our algorithm identifies and categorizes chart elements in a given view and adaptively performs a set of generalizations that selects and abstracts the elements based on a set of spatial and semantic constraints. The constraints determine the type of generalization operation that is applied to the resized line chart to minimize visual clutter. We implemented our algorithm using HTML5/JavaScript and D3 [10]. We now discuss the algorithm in more detail.

3.1 Identifying Semantic Importance

The first step in our algorithm is to identify all of the elements in the given chart. The algorithm assigns higher semantic weights to local extrema, first and last data values compared to chart elements such as axes or tick marks. Each element is assigned a unique ID and the algorithm computes a bounding box for each of them. Similar elements are then categorized into layers. This allows us to apply different constraints and generalization operators based upon the unique characteristics and semantics of each element layer. Each layer is assigned a value from 0 (lowest importance) to 1 (highest importance) based on the semantics of a line chart [27].

3.2 Compute Spatial Metrics

The algorithm then computes various spatial metrics for the elements in the resized line chart to minimize visual clutter. Based on visualization best practices [29, 14, 40], the amount of visual clutter is informed by the following guidelines:

Avoid congestion: A visualization view should not include too many elements at specific region.

Avoid conflict: To maintain legibility, elements should not overlap. Each element should be easily identifiable and readable.

Make more semantically important elements prominent: An element of higher importance value should be more visible than less important ones.

We now describe each of these spatial metrics in more detail.
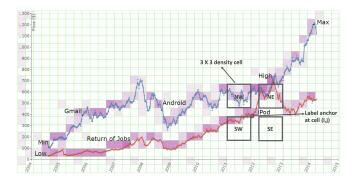
3.2.1 Density

Figure 2: Computing density. The entire multivariate line chart view is divided into $n \times m$ uniform cells and density at each cell is calculated to determine the number of elements visible within each cell area. Darker the square, higher the density.

Data density can provide simple, yet effective metrics for evaluating visual clutter in the view [40, 32]. To calculate density, we divide the entire visualization view into uniform $n \times m$ cells (where m and n are empirically determined based on the target display size). We then apply Topfer's Radical Law [38, 44] to express the number of elements that can be maintained at that size (Figure 2):

$$\text{Cell information density} = \frac{\text{Number of Elements}}{\text{Number of Pixels in the Cell}} \quad (1)$$

This metric allows for generalization operators to be applied to different regions based on their density value. For example, we can show more information or enlarge elements in less dense regions.

3.2.2 Distance

In our algorithm, the distance is measured between similar elements within the same layer such as labels or tick marks. The distance is measured using Euclidean distance. We utilize this metric to assess whether elements (e.g., annotations and associated data points) are too close to each other.

3.2.3 Collision

This metric determines collisions between elements in a chart that contribute to visual clutter. We employ a quadtree, a compact data structure to keep the algorithm performant [26]. We use the following equation to compute the overlapping area between elements:

$$\text{Collision area} = \sum_{i}^{N} \sum_{j \neq i}^{N} A(i,j) \tag{2}$$

where $A(i,j)$ is the area of overlap between elements i and j.

3.2.4 Area ratio

As the display size decreases, the proportion of areas that elements occupy with respect to the area of the resized chart, increases. This metric assesses the ratio of the total area of elements to the area of the entire visualization view and is used to maintain the area of the more semantically important elements in the chart, deemphasizing the area of less important ones. Area ratio is computed as follows:

$$\text{Area ratio} = \frac{\text{Area of element}}{\text{Total display area}} \tag{3}$$

3.3 Generalization operators

The primary goal of generalization is to maintain the recognizability of important elements, while deemphasizing less important information. We describe four generalization operations based on the spatial metrics - jittering, elimination, simplification, and merging.

3.3.1 Jittering

Jittering is a technique used to resolve collisions between elements by displacing the elements from their original positions to reduce visual clutter. We employ and extend the label placement simulated annealing algorithm [41] by adding additional heuristics:

1. A label should not overlap a data point.
2. A label should be located at the place which has the lowest information density.
3. A label's text-anchor should be updated based on the new position of the label.

We first determine a 3×3 density cell diagonally adjacent to a label anchor located in cell (i, j) with density (Figure 2). We chose a 3×3 cell neighborhood as it provided a reasonable heuristic for determining visual clutter. The sum of each 3×3 density cells is computed as follows:

$$
\begin{aligned}
\text{Northwest (NW)} &= d(i-3..i-1, j-3..j-1)\\
\text{Northeast (NE)} &= d(i+1..i+3, j-3..j-1)\\
\text{Southwest (SW)} &= d(i-3..i-1, j+1..j+3)\\
\text{Southeast (SE)} &= d(i+1..i+3, j+1..j+3)
\end{aligned}
\tag{4}
$$

where $d(a..b,c..d)$ denotes the sum of cell information density $d(x,y)$ for all $a \leq x \leq b$ and $c \leq y \leq d$. i and j are the indices of cells where the anchor is located and the screen coordinates where the origin is, at the top-left. After calculating the sum of each of the four directions, jittering is applied by calculating a new position that has the minimum density sum: $min(\text{NW}, \text{NE}, \text{SW}, \text{SE})$

3.3.2 Elimination

There is a high likelihood that too many elements will occupy a small area when the scale is reduced. This situation significantly increases both local information density and the likelihood of conflict. We compute a score S based on semantic importance for each element to determine which of the elements ought to be removed.

Figure 3: Left: Before elimination. Right: After elimination. Here, a few intermediate labels are removed as they are less important than other labels denoting start and end points as well as local maxima.

$$S = (1 - \text{imp}) \times W_{imp} + \text{local density} \times W_{dens} + \text{overlap} \times W_{ov} \tag{5}$$

where W_{imp}, W_{dens}, and W_{ov} are weights for the importance *imp*, local density, and overlap constraints respectively. Algorithm 1 performs the jittering and elimination operations and Figure 3 shows how elimination is applied to remove less important labels.

Algorithm 1 Elimination operation

Input: Resized line chart without elimination applied and semantic importance score S for all chart elements.
Output: Resized line chart with elimination applied.

1: **while** ! satisfyConstraints **do**
2:     Perform jittering to ascertain if constraints are satisfied.
3:     Eliminate elements starting with lowest S that overlap the most with other elements in the chart.
4: **end while**

3.3.3 Simplification

To reduce this complexity and clarify semantic intent, we simplify the line using the Douglas-Peucker algorithm [17] line simplification method. We found that this line simplification method offers efficient compression ratios while retaining important visual features in the lines. As shown in Figure 1, simplification maintains the overall data shape, while preserving visually prominent features.
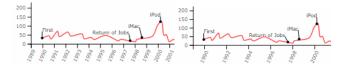
3.3.4 Merging

Figure 4: Left: Before merging. Right: After merging. Here, the domain of the x-axis is updated to merge yearly tick marks to two-year intervals.

This operation combines elements when there is congestion or the elements are in conflict. In the context of line charts, this operation applies to tick labels. As shown in Figure 4, two tick labels are merged into a single tick label by updating the axis domain and the interval between the tick marks.

3.4 Performance and space complexity

With our generalization algorithm, a new chart can be rendered on the fly based on the spatial constraints of the current view. While the overall complexity of the algorithm depends on the actual generalization operators employed, the complexity is at most $O(n^2)$ based

on the worst-case time-complexity of the jittering algorithm. The algorithm has been tested on a desktop browser for various target sizes and further investigation would need to be done to assess the algorithm's performance on mobile devices and smart watches.

4 EVALUATION

We conducted an evaluation of the algorithm with the following goals: (1) collect qualitative feedback on the usefulness of the generalization algorithm on various target display sizes and (2) identify limitations and future opportunities. Because the main goal of our study was to gain qualitative insight in the algorithm behavior, we encouraged participants to think aloud with the experimenter.

4.1 Method

4.1.1 Participants

We recruited 30 volunteers (12 males, 18 females, age 24 – 54) from a local town mailing list. The participants had a variety of backgrounds - software engineer, technical writer, sales consultant, product manager, program manager, and graduate student. Based on self-reporting, all were fluent in English and had experience reading line charts. 22 used a visualization tool [2, 1] on a regular basis and the rest considered themselves having limited proficiency.

4.1.2 Procedure and Apparatus

We ran a between-subjects design study where each participant was randomly assigned one resized chart from a set of 10 single-line charts. We sourced the charts from Pew Research [31], Wikipedia [42], and Tableau Public [37] and recreated them in D3. We removed graphical elements that could potentially affect the readability of the features such as highlighting and background shading. All the original charts had a display size of 6307×3220.

Task 1: During the first part of the study, each participant was initially shown the original line chart and then randomly shown one of three resized images of the chart, targeted at tablet (1536×2048), phone (750×1334), and watch (324×394) display sizes. The display sizes were chosen based on evaluation criteria commonly used to evaluate visualizations on mobile devices [9]. Because we were seeking subjective responses, participants could complete only one trial to avoid biases that might arise from repeated exposure to the task. To assess whether features in the chart were discernible, participants were asked the values for the start and end, local extrema, minimum, and maximum data points. We did not impose a constraint on the amount of time spent looking at the chart and answering the questions, emulating chart reading in the real world.

Task 2: The second part of the study was exploratory where participants used the prototype with the original line chart loaded into a web browser. They could resize the chart to any arbitrary display size by dragging and resizing the browser window. The study concluded with an interview.

The prototype was hosted on the experimenter's 16-inch laptop, a 2.4 GHz MacBook Pro running macOS Catalina 10.15.7 set to a resolution of 3072×1920. Each session took about 45 minutes.

4.1.3 Analysis Approach

We employed a mixed-methods approach involving qualitative and quantitative analysis, but considered the quantitative analysis mainly as a complement to our qualitative findings.

5 RESULTS AND DISCUSSION

Overall, participants were positive about the system and identified many benefits for the system being able to dynamically generalize the line chart to different display sizes. Several participants were impressed with the system's ability to preserve local extrema features and the general data shape in the line chart at smaller display sizes - ("I can recognize the smaller chart and has kept the peaks that I observed in the larger one (*P*2)."). All the participants were

able to identify the start and end values in the three generalized line charts during Task 1. 100% for the larger (1536×2048) size, 80% (8/10) for the phone (750×1334) size, and 30% (3/10) of the participants for the watch (324×394) size were able to accurately provide values for the local extrema, minium, and maximum points. Participants appreciated the sparkline appearance in the smallest chart and commented that such a representation would be useful for certain applications. For example, "I go running and this chart can be useful to show my heart rate on my iWatch. (*P*25)" and "I see these charts a lot for stock prices. They are simple and easy to read at a quick glance (*P*14)." During the exploratory part of the study (Task 2), participants appreciated the dynamic nature of the algorithm to be able to generalize the chart as they resized the browser window. Comments related to this behavior included, "I can see this being very useful if I'm switching between devices and want to view the same data (*P*2)" and "I often need to add charts into my table. It would be neat if I can drag a line chart into a cell and it creates a little microchart for me (*P*10)."

The study has limitations and provides opportunities for further improving and expanding the scope of the generalization behavior. **Exploring complex chart types and tasks**: We evaluated our generalization algorithm for univariate line charts. While the spatial metrics apply to multivariate line charts, there are interesting research opportunities to explore how well generalization applies to other chart types. *P*4 said, "I use a lot of Excel tables, but often struggle to get the gist of them when I view them on my mobile device." Exploring how data distributions can affect the spatial metrics and semantic properties of chart elements, is another interesting direction of research inquiry. The task of reading data values from the charts at different scales, is rather simplistic. Future work should explore more complex chart reading tasks including trends, seasonal patterns, and overall takeaways.

Control over feature retention and simplification: A challenge for automating the generalization process is representing data at multiple scales [13]. Participants wanted to have more control over what information is preserved as the line chart is resized. *P*7 commented, "I wanted to keep the label on the small dip in the dollar price as I felt that was important and I wish there was a way to keep it sticky." Similar to responsive web design tools [20] that support both automation and flexibility of content and layout, there is a need to explore responsive chart and dashboard authoring tools.

Generalization with deep learning models: Our approach employs spatial metrics for determining the generalization operation. Neural networks and deep learning methods are promising avenues for learning good generalization examples through pattern recognition [23]. Exploring how such deep learning approaches can replicate many difficult aspects of expert generalization, such as line smoothing, enlargement, and displacement, would help scale the applicability of chart generalization over a variety of use cases.

Finally, future work should test these ideas in a live system for specific real-world analytical tasks and actual target display devices rather than in the artificial setting of a study.

6 CONCLUSION

This paper presents a technique for applying cartographic principles to the resizing of line charts at different target display sizes. Specifically, we present a set of spatial metrics to examine geometric properties and relationships among elements in line charts. We employ these metrics to determine a set of generalization operations for semantically resizing line charts; maintaining the legibility of visually prominent features and demphasizing less important features. An evaluation of the algorithm indicates that participants found that the generalized line charts preserved the general data shape and features from that of the original. Feedback from interacting with the prototype identified opportunities for further exploring the space of generalization techniques for creating responsive chart design.

REFERENCES

[1] Microsoft Q&A. https://powerbi.microsoft.com, 2021.

[2] Tableau Software. https://tableau.com, 2021.

[3] M. Agrawala and C. Stolte. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249. ACM, 2001.

[4] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007.

[5] S. Baldassi, N. Megna, and D. C. Burr. Visual clutter causes high-magnitude errors. *PLoS biology*, 4(3):e56, 2006.

[6] R. Ball, C. North, and D. A. Bowman. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 191–200. ACM, 2007.

[7] B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST '94, pages 17–26, New York, NY, USA, 1994. ACM.

[8] T. Blascheck, L. Besançon, A. Bezerianos, B. Lee, and P. Isenberg. Glanceable visualization: Studies of data comparison performance on smartwatches. *IEEE transactions on visualization and computer graphics*, 25(1):630–640, 2018.

[9] K. Blumenstein, C. Niederer, M. Wagner, G. Schmiedl, A. Rind, and W. Aigner. Evaluating information visualization on mobile devices: Gaps and challenges in the empirical evaluation design space. In *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*, BELIV '16, page 125–132, New York, NY, USA, 2016. Association for Computing Machinery.

[10] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[11] K. E. Brassel and R. Weibel. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information System*, 2(3):229–244, 1988.

[12] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe. Visualizing ranges over time on mobile phones: a task-based crowdsourced evaluation. *IEEE transactions on visualization and computer graphics*, 25(1):619–629, 2018.

[13] B. Buttenfield, R. McMaster, and H. Freeman. *Map Generalization: Making Rules for Knowledge Representation*. New York, NY : Wiley, 1991.

[14] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[15] Y. Chen, W.-Y. Ma, and H.-J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 225–233, New York, NY, USA, 2003. ACM.

[16] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1), Jan. 2009.

[17] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[18] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, and M. Schraefel. Touchviz: a case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2301–2310, 2013.

[19] S. G. Eick, J. L. Steffen, and E. E. Sumner, Jr. Seesoft-a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, Nov. 1992.

[20] B. Gremillion. *Responsive Web Design: Getting the New Baseline in Web Design Right*. Smashing Magazine, 2013.

[21] J. Hoffswell, W. Li, and Z. Liu. Techniques for flexible responsive visualization design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[22] T. Horak, A. Mathisen, C. N. Klokmose, R. Dachselt, and N. Elmqvist. *Vistribute: Distributing Interactive Visualizations in Dynamic Multi-Device Setups*, page 1–13. Association for Computing Machinery, New York, NY, USA, 2019.

[23] I. Karsznia and K. Sielicka. When traditional selection fails: How to improve settlement selection for small-scale maps using machine learning. *ISPRS International Journal of Geo-Information*, 9:230, 04 2020.

[24] H. Kim, D. Moritz, and J. Hullman. Design patterns and trade-offs in responsive visualization for communication, 2021.

[25] H. Kim, H. Oh, Y. Han, S. Ko, M. Brehmer, and B. C. Kwon. Thumbnails for data stories: A survey of current practices. In *2019 IEEE Visualization Conference (VIS)*, pages 116–120, 2019.

[26] A. Klinger. Patterns and search statistics. pages 303–337, 1971.

[27] S. M. Kosslyn. Understanding charts and graphs. *Applied cognitive psychology*, 3(3):185–225, 1989.

[28] C. Kray, C. Elting, K. Laakso, and V. Coors. Presenting route instructions on mobile devices. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, pages 117–124, New York, NY, USA, 2003. ACM.

[29] A. M. MacEachren. *How Maps Work - Representation, Visualization, and Design*. Guilford Press, 2004.

[30] R. B. McMaster and K. S. Shea. Generalization in digital cartography. Association of American Geographers Washington, DC, 1992.

[31] Pew Research. https://www.pewresearch.org, 2021.

[32] R. Rosenholtz, Y. Li, and L. Nakano. Measuring visual clutter. *Journal of Vision*, 7(2):17, 2007.

[33] V. Roto, A. Popescu, A. Koivisto, and E. Vartiainen. Minimap: a web page visualization method for mobile phones. In *Proceedings of the SIGCHI conference*, pages 35–44, NY, USA, 2006. ACM Press.

[34] M. Schwab, S. Hao, O. Vitek, J. Tompkin, J. Huang, and M. A. Borkin. *Evaluating Pan and Zoom Timelines and Sliders*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2019.

[35] K. S. Shea and R. B. McMaster. Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings of AutoCarto*, volume 9, pages 56–67, 1989.

[36] W. Shi and C. Cheung. Performance evaluation of line simplification algorithms for vector generalization. *The Cartographic Journal*, 43(1):27–44, 2006.

[37] Tableau Public. https://public.tableau.com, 2021.

[38] F. Töpfer and W. Pillewizer. The principles of selection. *The Cartographic Journal*, 3(1):10–16, 1966.

[39] J. Trevor, D. M. Hilbert, B. N. Schilit, and T. K. Koh. From Desktop to Phonetop: A UI for web interaction on very small devices. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 121–130, New York, NY, USA, 2001. ACM.

[40] E. R. Tufte and P. Graves-Morris. *The Visual Display of Quantitative Information*, volume 2. Graphics press Cheshire, CT, 1983.

[41] E. Wang, G. G. Park, and Yeoli. A D3 plug-in for automatic label placement using simulated annealing. 2013.

[42] Wikipedia. https://www.wikipedia.org, 2021.

[43] L. Wolf, M. Guttmann, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6. IEEE, 2007.

[44] A. Woodruff, J. Landay, and M. Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 19–28. ACM, 1998.

[45] A. Wu, W. Tong, T. Dwyer, B. Lee, P. Isenberg, and H. Qu. Mobilevisfixer: Tailoring web visualizations for mobile phones leveraging an explainable reinforcement learning framework. *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[46] Y. Wu, X. Liu, S. Liu, and K.-L. Ma. Visizer: a visualization resizing framework. *Visualization and Computer Graphics, IEEE Transactions on*, 19(2):278–290, 2013.

[47] A. Zipf. User-adaptive maps for location-based services (lbs) for tourism. In *9th Int. Conf. for Information and Communication Technologies in Tourism (ENTER 2002)*, pages 329–337, 2002.