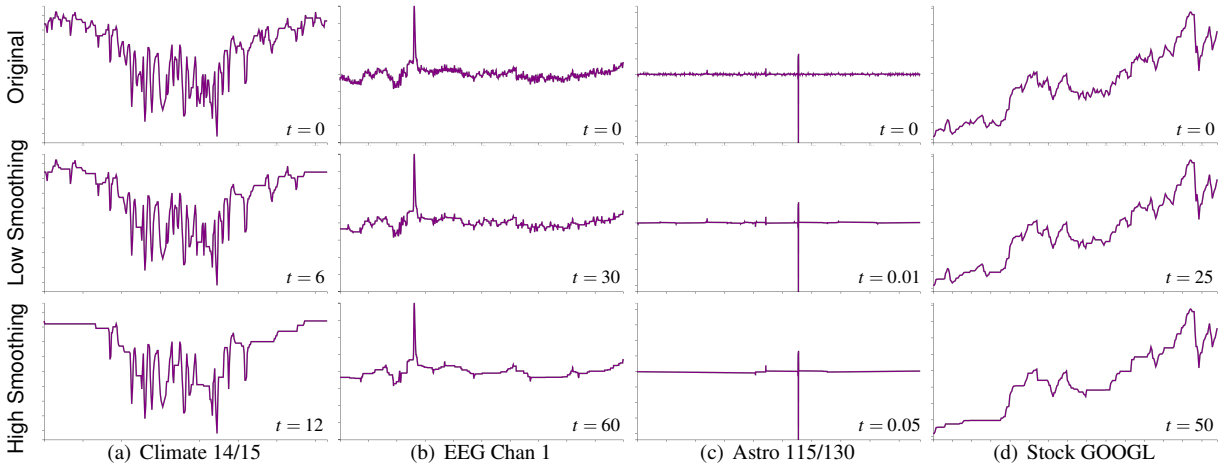
arXiv:1906.09457v1 [cs.GR] 22 Jun 2019

TopoLines: Topological Smoothing for Line Charts

Ashley Suh* Tufts University
Christopher Salgado† University of South Florida
Mustafa Hajij‡ Ohio State University
Paul Rosen§ University of South Florida



(a) Climate 14/15 (b) EEG Chan 1 (c) Astro 115/130 (d) Stock GOOGL

Figure 1: Examples from the 4 datasets (top) with low (middle) and high (bottom) levels of smoothing, t, using TopoLines. TopoLines works by specifically preserving high amplitude peaks, as defined by persistent homology, and flattening low amplitude peaks in a way that produces low residual error.

ABSTRACT

Line charts are commonly used to visualize a series of data samples. When the data are noisy, smoothing is applied to make the signal more apparent. However, the difference between signal and noise is ill-defined, as it depends upon the application domain of the data. Common methods used to smooth line charts, such as rank filters, convolutional filters, frequency domain filters, and subsampling, optimize on various properties of the data. However, these methods are only ideal in certain limited scenarios. We present TopoLines, a method for smoothing line charts by leveraging techniques from Topological Data Analysis. The design goal of TopoLines is to smooth line charts while maintaining prominent peaks, as defined by persistent homology, in the data. We compare TopoLines to 5 popular line smoothing methods using data from 4 application domains. We evaluate TopoLines in terms of l^2-norm of the residual as the simplification threshold is varied, and we perform a user study to evaluate users' perception of the accuracy of TopoLines.

Keywords: Line chart, data smoothing, Topological Data Analysis.

1 INTRODUCTION

Line charts are used in a variety of applications, such as analyzing stock trends, tracking weather changes, understanding brain activity, and understanding the chemical composition of distance stars. While significant increases in data availability allow users to create plots with many data points, relieving visual clutter requires performing additional data processing, such as smoothing.

*e-mail: ashley.suh@tufts.edu
†e-mail: csalgado1@mail.usf.edu
‡e-mail: hajij.1@osu.edu
§e-mail: prosen@usf.edu

A wide variety of smoothing techniques are available, each optimizing on different properties. One naïve approach is uniform subsampling, which skips data on a regular interval. While trivial to implement, the output of the approach is unstable as the interval is varied—it does nothing in particular to preserve any quality of the input data. Other commonly used methods, such as median, Gaussian, and low-pass filters, provide smoothing that retains low frequency aspects of signals but lose potentially important peaks in the data. Irregular sampling, such as Douglas-Peucker [8, 17], does a better job preserving peaks, but it retains very little signal detail in the smoothing process.

We address the weaknesses of prior approaches by applying Topological Data Analysis to line chart smoothing by specifically retaining prominent peaks in the data. We do this by using persistent homology to capture the hierarchical relationship between peaks (i.e., critical points in the form of local minima and maxima) in the line chart. We then apply topological smoothing to the line chart to remove critical points of low importance, while retaining as much of the original signal detail as possible.

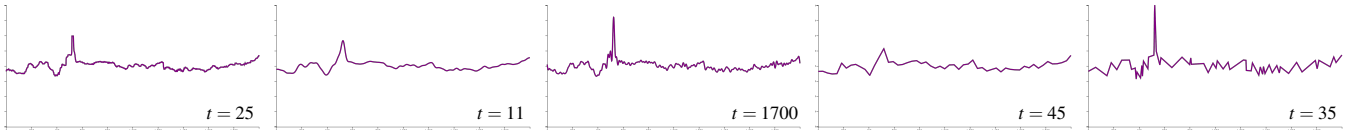
We show the benefits of TopoLines in 2 ways. First, we demonstrate that TopoLines produces predictable results by examining the l^2-norm of the residual produced by the smoothing. We show that the simplification level of TopoLines is monotonic and linearly correlated with l^2-norm of the residual, while many other approaches are not. Second, we demonstrate the preference for TopoLines by performing an Amazon Mechanical Turk user study comparing 4 dataset types and 5 existing smoothing techniques.

2 BACKGROUND: SMOOTHING LINE CHARTS

We discuss 5 classes of smoothing used in line charts. Each offers an adjustable simplification parameter, whose interpretation is approach dependent, and preserves particular properties of the input.

Rank Filters

Rank filters are nonlinear filters that sort values within a neighborhood window surrounding each input point and select a single value from that set for output. The *median filter* (see



(a) Median (b) Gaussian (c) Low-Pass (d) Uniform Subsampling (e) Douglas-Peucker

Figure 2: Output of standard methods with a *high-level* of smoothing on EEG Chan 1 data from Fig. 1(b). See supplementary material for all data.

Fig. 2(a)) selects the median value. The level of smoothing can be increased or decreased by enlarging or shrinking the window, respectively. Median filters are particularly good at removing salt-and-pepper noise [4]. However, these are exactly the types of peaks we are trying to retain with TopoLines. From a computational complexity point-of-view, repeated sorting operations are required, one per output point, making the overall performance $\mathcal{O}(n \cdot w \log w)$, where n is the number of data points and w is the window size.

Convolutional Filters Convolutional filters are a stencil-based method, where a series of weights are applied to an input neighborhood. The *Gaussian filter* (see Fig. 2(b)) is commonly used in convolutional signal and image processing [15]. It weights the input neighborhood using a normal distribution. Smoothing level is increased or decreased by adjusting the standard deviation, σ, of the distribution. The Gaussian filter can be seen as a form of low-pass filter, blurring both signal and noise from the data, producing smooth, visually appealing results. Usually the size of the window used for the Gaussian is fixed using σ as a guide. In our implementation, a window size of $\pm 3\sigma$ ensures that we capture 99.7% of the distribution. The resulting computational complexity is $\mathcal{O}(n \cdot \sigma)$.

Frequency Domain Filters Frequency domain filtering converts the scalar data into a frequency domain representation, via wavelets or Fourier transform. We compare to a *low-pass filter* (see Fig. 2(c)). Our low-pass filter converts the input into the frequency domain using a Discrete Fourier Transform (DFT) [6]. Zeroing out high frequency components smooths the output, increasing the level of smoothing by increasing the range of components that are zeroed out. Finally, the output is computed by converting the frequency domain data back to the spatial domain using an inverse DFT. Much like Gaussian filters, low-pass filters produce smooth output, in this case only retaining the specified frequencies. However, the relationship between the frequency and spatial domains is often not intuitive, as multiple frequency contribute to a single output. The computation of the DFT is $\mathcal{O}(n \log n)$, which is also the overall complexity of the low-pass filter.

Uniform Subsampling Subsampling approaches take the original data and select a subset of data points. A common choice, due to its ease of implementation, *uniform subsampling* (see Fig. 2(d)) selects points at regular intervals. Between selected points, interpolation is used, linear interpolation being the easiest case. Simplification is increased by sampling fewer points. Uniform subsampling makes few guarantees about the types of features it will preserve, unless the input is already *significantly over-sampled*, in which case it will retain the original signal. Computationally, uniform subsampling is very efficient taking only $\mathcal{O}(s)$, where s is the number of samples taken from the input.

Nonuniform Subsampling Nonuniform subsampling selects points at irregular intervals. *Douglas-Peucker* [8, 17] (see Fig. 2(e)) is an example that establishes a priority queue of points by optimizing l^∞-norm of the residual error (i.e., the difference between the original and smoothed line charts). The algorithm starts by selecting the boundary points of the input data (i.e., first and last points) for initialization and connects them via linear interpolation. Points are then iteratively added by selecting the input point with the largest distance from the current output and inserting it into the output. The process continues until a user specified threshold distance is reached. The simplification is increased or decreased by modifying this threshold. The output captured by Douglas-Peucker is reliable and predictable, in that the output will deviate no more than the specified threshold. However, it does not guarantee peaks are preserved nor does it retain additional details from the input. The worst case complexity of the algorithm is $\mathcal{O}(n^2)$, while average complexity is $\mathcal{O}(n \log n)$.

3 TopoLines: Topologically Smoothed Line Charts

TopoLines takes cues from a number of these approaches. It requires 2 main steps. In the first step, the topology of the signal is extracted and classified using persistent homology [10]. In the second step, the output signal is smoothed by removing critical points based upon a user-selected threshold.

3.1 Persistent Homology of the Line Chart

Our approach begins by extracting the topological features of a line chart by tracking the evolution of the H_0 homology groups, i.e., the connected components, from the data using persistent homology. We try to provide a practical description of persistent homology, leaving further details and theoretical justifications to [10].

We use the lower-star filtration of the simplicial complex, $\mathscr{F}$ (i.e., the points and edges), on the function $f : \mathscr{F} \to \mathbb{R}$. The lower-star filtration of the data tracks the creation and merging of connected components of the sublevelset $|\mathscr{F}|_i = f^{-1}(-\infty, f_i]$, as f_i is swept from $-\infty \to \infty$. To calculate this, data points are sorted by f in increasing order. Points are then inserted into $|\mathscr{F}|$ one at a time, and if their neighboring points are already in $|\mathscr{F}|_i$, an edge is added. Finally, connected components are extracted efficiently using the disjoint-set data structure.

The relationship between connected components is tracked using a merge tree. When a component first appears at f_i, caused by a local minimum, a new leaf node is created in the merge tree, also at f_i. For example in Fig. 3(c), the orange connected component is formed at Ⓔ, and an equivalent leaf node is created in the merge tree. As the plane is swept higher, as in Fig. 3(d), new connected components—Ⓐ in yellow and Ⓒ in green—are created.

When 2 components merge, representing a local maximum, a merge node is created in the merge tree at f_i and connected to the merged components. In Fig. 3(e)/3(f), the green and orange connected components merge at Ⓓ, a local maximum of f. At that point, the connected components are combined and a merge node is added to the merge tree.

When a merge node is created, the merge node is paired with a leaf node (i.e., a local maximum is paired with a local minimum). In particular, it is paired with the minimum from the two merging components with the *larger* value. Referring again to Fig. 3(e)/3(f), the point Ⓓ is paired with the minimum from the green and orange components with the larger value, in this case point Ⓒ. In other words, $f(Ⓒ) > f(Ⓔ)$, therefore, [Ⓒ, Ⓓ) form a feature pair. The new merged component in orange continues on with minimum Ⓔ. Similarly, in Fig. 3(f)/3(g), at Ⓑ, the value of the minimum of yellow $f(Ⓐ)$ and orange $f(Ⓔ)$ are compared, and [Ⓐ, Ⓑ) are paired.

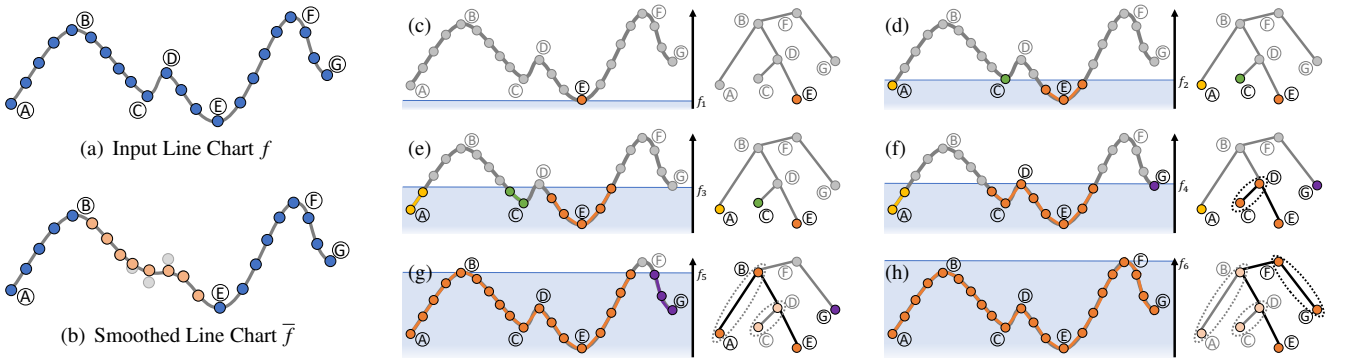


Figure 3: (a-b): Example line chart before and after smoothing one feature pair. (c-h): Lower-star filtration (left) and merge tree construction (right).

The output from this operation is the set of all feature pairs, $C = \{[b_0,d_0),[b_1,d_1),...,[b_m,d_m)\}$, where b_i and d_i are the local minimum and maximum, respectively. For *line charts*, it can be shown that this algorithm takes $\mathcal{O}(n+m\log m)$[1]. Otherwise, as described, it takes $\mathcal{O}(n\log n)$.

3.2 Topological Simplification

The set of critical point pairs, C, is used to guide smoothing, as follows. For each pair, a measure known as *persistence* is calculated. Persistence is simply the difference in function value between the local minimum and local maximum of the pair, in other words, $p_i = |f(d_i) - f(b_i)|$. In effect this measures the *peak-to-peak amplitude*.

A user specified threshold, t, provides the level of simplification by selecting all critical point pairs whose persistence is less than t, that is, $\{C_i | p_i < t\}$, for removal from the output.

Using critical point pairs to simplify a scalar function has been studied extensively in the context of 2D manifolds [11] and contour trees in general [5]. The observation behind these approaches is that in order to remove a pair of critical points from the function, one simply needs to "flatten" the area surrounding them. We adapt this observation to our context.

In the context of a 1D function, "flattening" equates to making the function strictly monotonic between the critical points immediately left and right of the selected critical point pair[2]. For example, in Fig. 3(a), removing the [Ⓒ,Ⓓ] critical point pair requires modifying the function such that it is monotonically decreasing between critical points Ⓑ and Ⓔ. The design space of possible modifications is quite large—*any* monotonically decreasing function satisfies the topological constraint of critical point removal. We apply the additional constrain that we would like to otherwise modify the remainder of the function as little as possible.

To perform this simplification, we operate on the derivative of the function, f' (see Fig. 4(a)). Here, f' is a dual representation of f—points in f are lines in f' and lines in f are points in f'. For functions to be strictly monotonic: $f'_i > 0$ for increasing functions, and $f'_i < 0$ decreasing functions. For the example pair of [Ⓒ,Ⓓ], the function needs to be made monotonically decreasing, that is, $f'_i < 0$ for $i \in (Ⓑ,Ⓔ)$. Practically, our approach does this by setting $\overline{f'_i} = f'_i$ if $f'_i < 0$ and $\overline{f'_i} = -\varepsilon$ otherwise, where ε is a small value.

Finally, in order to produce the correct output, the total derivative needs to remain the same, such that $\sum f' - \sum \overline{f'_i} = 0$. To do this, any excess in the derivative is propagated equally to the left and right of the critical point pair [Ⓒ,Ⓓ], while keeping $\overline{f'_i} < 0$.

Finally, the output curve is calculated by integrating $\overline{f'}$ into $\overline{f}$. In Fig. 4(a), the derivative f' is calculated for the input, Fig. 3(a). The orange point, which has a value greater than zero, is decreased because of the monotonicity constraint. The excess difference between derivatives f' and $\overline{f'_i}$ is propagated to the blue points to the left and right of [Ⓒ,Ⓓ], resulting in the final $\overline{f'_i}$ in Fig. 4(b). Finally, $\overline{f'_i}$ is integrated to produce the output curve in Fig. 3(b).

The computational complexity for smoothing *each* pair is linear. Naïvely implemented, the worst case overall complexity is $\mathcal{O}(n^2)$. However, the computation can be made non-redundant using properties of the merge tree, improving complexity to $\mathcal{O}(n)$.

[1]This is done by sweeping the input data and removing noncritical points.
[2]Pairs with critical points on the boundary use the boundary point instead.

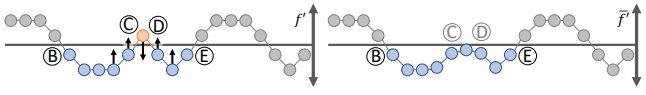


Figure 4: TopoLines smoothing using the derivative of the function.

4 Results

To evaluate the effectiveness of TopoLines, we compare it to 5 other smoothing methods: median filter, Gaussian filter, low-pass filter, uniform subsampling, and Douglas-Peucker (D-P).

We test our method using 4 application domains: *weather*, electroencephalogram (*EEG*), radio *astro*nomy, and *stock* trends. For each domain, we tested 4 datasets. *Weather* is a measure of daily high temperature recorded July to July over 4 periods (20-14/15, 15/16, 16/17, and 17/18) at a large metropolitan area airport downloaded from [27]. *EEG* data contains a small window of 4 (of 32 total) channels from a single subject undergoing a visual attention task and was acquired from [7]. Radio *astro*nomy data are 4 spectral "lines" that measure the frequency and amplitude of radio waves emitted by extraterrestrial matter (i.e., gas and dust) and were downloaded from [1]. *Stock* trends contain daily closing values for 4 technology companies (Amazon, Google, Intel, and Microsoft) over an 8 month period starting in July 2017, collected from Google Finance. Examples for each domain, along with TopoLines smoothing, can be found in Fig. 1.

Source code is available at <*withheld for double blind review*>. The complete results for all data and smoothing methods can be found in our supplementary materials.

4.1 Residual Error Comparisons

The smoothing thresholds used on each of the 6 methods (5 above plus TopoLines) have almost no relationship to one another, making direct comparison difficult. Instead, we compare properties of the residual error, as measured by the l^2-norm, produced by each smoothing method as the threshold is varied. For a given function, f, and a smoothed version of the function, $\overline{f_t}$, with smoothing threshold t,

the l^2-norm of the residual is calculated as $l_t^2 = \sqrt{\sum (f(x) - \overline{f_t}(x))^2}$. In our experiments the range of the smoothing threshold t was manually calibrated by identifying the "elbow" of the l_t^2 function of the residual for each approach and dataset.

We evaluate the relationship between t and l_t^2 in 2 regards[3]. First, l_t^2 should always be monotonically increasing with respect to t. In other words, as the smoothing threshold goes up, so should the residual error. Second, l_t^2 should be linearly correlated to t (using Pearson Correlation Coefficient), meaning that as the threshold is modified, the level of error is predictable, in a linear sense.

Table 1 shows the results. Monotonicity is categorized by monotonic, approximately monotonic (containing a drop of no more than 5% of the maximum), and non-monotonic. The results show that for our data *TopoLines, along with Gaussian and low-pass filters, are monotonic*, while D-P, median, and subsampling are not. Correlation is categorized by ranges of mostly correlated $(0.85, 1]$, somewhat correlated $(0.7, 8.5]$, and other $[-1, 0.7)$. The results show that for our data TopoLines and D-P are mostly correlated; Gaussian and low-pass are usually but not always correlated; and median and subsampling can be unreliable.

Table 1: Evaluation of the relationship between filter level and l^2-norm of the residual, measures monotonicity and correlation coefficient.

Dataset	D-P		Gauss.		Low-P.		Median		Subsamp		TopoLines	
Climate 14/15	N	0.98	M	0.90	M	0.95	N	0.81	N	0.86	M	0.99
Climate 15/16	N	0.96	M	0.93	M	0.95	N	0.82	N	0.91	M	0.99
Climate 16/17	N	0.99	M	0.93	M	0.95	N	0.84	N	0.94	M	0.98
Climate 17/18	A	0.98	M	0.96	M	0.93	N	0.84	N	0.95	M	0.97
EEG Chan 1	A	1.00	M	0.93	M	0.80	N	0.98	N	0.88	M	0.99
EEG Chan 7	A	1.00	M	0.90	M	0.84	N	0.87	N	0.88	M	1.00
EEG Chan 14	A	0.99	M	0.85	M	0.81	N	0.83	N	0.82	M	0.99
EEG Chan 21	M	0.99	M	0.97	M	0.68	N	0.93	A	0.97	M	0.98
Astro 115/130	A	0.91	M	0.73	M	0.99	M	-0.96	N	0.15	M	0.94
Astro 116/133	N	0.91	M	0.90	M	0.99	A	0.77	N	0.56	M	0.87
Astro 117/121	A	0.98	M	0.93	M	1.00	N	0.62	N	0.7	M	0.93
Astro 119/123	M	0.93	M	0.97	M	0.95	N	0.61	N	0.85	M	0.93
Stock AMZN	A	0.99	M	0.97	M	0.93	N	0.94	N	0.96	M	0.95
Stock GOOGL	A	0.99	M	0.99	M	0.93	N	0.97	N	0.96	M	0.92
Stock INTC	M	0.93	M	0.99	M	0.91	N	0.98	N	0.95	M	0.96
Stock MSFT	N	0.99	M	0.97	M	0.94	N	0.94	N	0.97	M	0.98

M: Monotonic; A: Approximately Monotonic; N: Non-Monotonic
Correlation Coefficient: $(0.85, 1.0]$; $(0.7, 0.85]$; $[-1.0, 0.7]$

4.2 User Preference Study

We performed a study using Amazon Mechanical Turk to determine user preference between smoothing techniques. The study consisted of a demographics questionnaire, 4 warm-up line chart questions, 50 test questions, and 1 post-test question. Users were compensated in accordance with US minimum wage.

[3]The plots of t vs. l_t^2 can be found in our supplementary material.

Table 2: Evaluation of user preference for TopoLines over alternative approaches (larger is better).

Dataset	D-P	Gauss.	Low-P.	Median	Subsamp
Climate 14/15	58% (n=31)	85% (n=20)	34% (n=38)	61% (n=33)	45% (n=22)
Climate 15/16	76% (n=29)	88% (n=26)	47% (n=32)	89% (n=35)	82% (n=33)
Climate 16/17	68% (n=25)	83% (n=29)	47% (n=43)	91% (n=23)	84% (n=31)
Climate 17/18	41% (n=34)	77% (n=22)	51% (n=39)	60% (n=25)	69% (n=29)
EEG Chan 1	54% (n=28)	92% (n=26)	23% (n=40)	75% (n=28)	88% (n=26)
EEG Chan 7	51% (n=43)	93% (n=29)	38% (n=26)	91% (n=33)	85% (n=34)
EEG Chan 14	61% (n=31)	86% (n=21)	34% (n=29)	61% (n=33)	79% (n=34)
EEG Chan 21	63% (n=43)	82% (n=28)	43% (n=37)	79% (n=34)	85% (n=27)
Astro 115/130	64% (n=36)	93% (n=42)	66% (n=47)	91% (n=34)	87% (n=30)
Astro 116/133	54% (n=35)	90% (n=30)	70% (n=37)	97% (n=35)	94% (n=32)
Astro 117/121	38% (n=24)	77% (n=26)	41% (n=29)	51% (n=37)	86% (n=28)
Astro 119/123	50% (n=20)	90% (n=30)	58% (n=36)	74% (n=43)	79% (n=33)
Stock AMZN	74% (n=35)	89% (n=47)	76% (n=33)	86% (n=35)	87% (n=47)
Stock GOOGL	67% (n=43)	87% (n=46)	76% (n=34)	92% (n=39)	100% (n=43)
Stock INTC	74% (n=23)	68% (n=28)	39% (n=44)	72% (n=29)	83% (n=29)
Stock MSFT	69% (n=45)	85% (n=39)	56% (n=45)	77% (n=39)	80% (n=44)

TopoLines Preference: $[60\%, 100\%]$; $(40\%, 60\%)$; $[0\%, 40\%]$

The preference questions consisted of a single task: "*Click on the plot that looks most similar to the plot of the original*". For this task, users were shown the original line chart at 400x200px resolution and 3 smoothed line charts at 300x150px resolution. The smoothing type and threshold were selected at random from the same ranges used in the previous section, except that 1 plot was always TopoLines. Users had 30 seconds to select a plot or an 'Unsure' button. An example is available in the supplemental material.

We recruited a total of $n = 35$ (12 female, 23 male) participants who completed the entire survey. The age of participants was reported as: 1–[18,24]; 22–[24,34]; 9–[35,44]; and 3–[55,64]. Asked about visualization usage: 2–None; 15–Minimal Use; 14–Casual Use; 3–Regular Use; and 1–Extensive Use.

We tested 35 participants x 50 questions for a total of 1750. 33 results were excluded due to time expiring or users selecting 'unsure' for a total of $n = 1717$ results used in the analysis. The complete results, raw and visualized, can be found in supplementary materials.

We compare user preference by looking at the frequency that users selected TopoLines instead of the alternative smoothing method. Table 2 shows a summary for all datasets and filter types. TopoLines almost always outperformed Gaussian, median, and subsampling. TopoLines also outperformed D-P, despite the results being more balanced. Finally, the results between TopoLines and low-pass filter were mixed.

5 Discussion

Based upon our analysis, TopoLines, low-pass filtering, and, to a slightly lesser extent, Douglas-Peucker all performed well. Even so, these methods have limitations, not necessarily accounted for in the evaluation. For example, low-pass filtering can develop "ringing" artifacts near the boundary, and determine the smoothing threshold in the frequency domain can be challenging. Nevertheless, the important relationship between all 3 approaches is that they tend to preserve large peaks in the data. Given this observation, TopoLines, whose primary objective is to optimize the preservation of high amplitude peaks, has an advantage. However, deeper analysis of user preference is required to fully optimize the result, particularly with respect to the large design space available with smoothing, since the only requirement is that the output becomes monotonic.

6 Prior Work on Perception of Line Charts

Although work has been done regarding topological smoothing [5, 11, 23], we could find no prior studies that compared smoothing techniques in line charts. However, there have been many studies that consider overall perception of line charts.

Prior research has investigated how line chart visualizations affect user preference and ability. Techniques that optimize for graphical perception [13,19,24] have been shown to improve both the accuracy and effectiveness for understanding line chart data [3, 12, 14]. For example, Javed et al. evaluated the effectiveness of line charts by testing users on their ability to perform comparison, slope, and discrimination tasks for each line chart technique. The results showed that, when comparing line charts using small multiples, horizon graphs, stacked graphs, and braided graphs, user performance varied depending on the data being displayed using separate or shared-space charts during analysis [14].

Recent work has also been done to evaluate the influence of aspect ratio [26], visual complexity [18], and user interaction [2] with line charts. Wang et al. provided a robust selection method to recommend aspect ratios in line charts using the resultant vector and local orientation resolution [26]. Ryan et al. proposed a technique to quantitatively measure the visual complexity in line charts using Pixel Approximate Entropy (PAE) [18]. Adnan et al. conducted graphical perception study to measure the effectiveness of interaction during the analysis of time series data [2].

Although line charts are a common method for visualizing one dimensional data [16], extensive studies have been completed to determine their effectiveness compared to other chart types. One study confirmed the tendency of users to associate bar charts with discrete entities and line charts with trends [28]. A more recent study explored this hypothesis further in a multi-chart experiment [20]. They found that line charts are significantly more accurate than other charts for correlation and distribution tasks, thus, line charts are best for tasks requiring finding correlations. Finally, Wang et al. provided a method to automatically select an appropriate chart (line chart vs. scatter plot) when visualizing time series data [25].

Other line chart visualization challenges that have been investigated include accounting for user performance when there is missing data [9, 21, 22]. For example, Song et al. demonstrated the different design choices that should be leveraged for line charts and bar graphs when estimating averages and trends that are missing values [21].

Acknowledgments

We would like to thank Bei Wang for providing valuable feedback on this project. This work was supported in part by a grant from National Science Foundation (IIS-1513616).

References

[1] Alma science archive. `http://almascience.nrao.edu/aq/`. Accessed: 2019-06-01.

[2] M. Adnan, M. Just, and L. Baillie. Investigating time series visualisations to improve the user experience. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5444–5455, 05 2016. doi: 10.1145/2858036.2858300

[3] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, Jan 2017. doi: 10.1109/TVCG.2016.2598592

[4] G. R. Arce. *Nonlinear signal processing: a statistical approach*. John Wiley & Sons, 2005.

[5] H. Carr, J. Snoeyink, and M. van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry: Theory and Applications*, 43(1):42–58, 2010.

[6] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[7] A. Delorme. Eeg / erp data available for free public download. `https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html`. Accessed: 2019-06-01.

[8] 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.

[9] C. Eaton, C. Plaisant, and T. Drizd. Visualizing missing data: Graph interpretation user study. In *IFIP Conference on Human-Computer Interaction*, pp. 861–872. Springer, 09 2005. doi: 10.1007/11555261_68

[10] H. Edelsbrunner and J. Harer. Persistent homology - a survey. *Contemporary Mathematics*, 453:257–282, 2008.

[11] H. Edelsbrunner, D. Morozov, and V. Pascucci. Persistence-sensitive simplification of functions on 2-manifolds. *Proceedings of the 22nd Annual ACM Symposium on Computational Geometry*, pp. 127–134, 2006.

[12] A. Gogolou, T. Tsandilas, T. Palpanas, and A. Bezerianos. Comparing similarity perception in time series visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):523–533, Jan 2019. doi: 10.1109/TVCG.2018.2865077

[13] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1303–1312, 04 2009. doi: 10.1145/1518701.1518897

[14] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, Nov 2010. doi: 10.1109/TVCG.2010.162

[15] S. K. Kopparapu and M. Satish. Identifying optimal gaussian filter for gaussian noise removal. In *2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pp. 126–129, Dec 2011. doi: 10.1109/NCVPRIPG.2011.34

[16] S. Kosslyn. Understanding charts and graphs. *Applied Cognitive Psychology*, 3:185 – 225, 07 1989. doi: 10.1002/acp.2350030302

[17] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.

[18] G. Ryan, A. Mosca, R. Chang, and E. Wu. At a glance: Pixel approximate entropy as a measure of line chart complexity. *IEEE transactions on visualization and computer graphics*, 25(1):872–881, 2019.

[19] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In *IEEE Symposium on Information Visualization (InfoVis)*, pp. 173–180. IEEE, 2005.

[20] B. Saket, A. Endert, and A. Demiralp. Task-based effectiveness of basic visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(7):2505–2512, July 2019. doi: 10.1109/TVCG.2018.2829750

[21] H. Song and D. Albers Szafir. Where's my data? evaluating visualizations with missing data. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 08 2018. doi: 10.1109/TVCG.2018.2864914

[22] M. Templ, A. Alfons, and P. Filzmoser. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*, 6:29–47, 04 2012. doi: 10.1007/s11634-011-0102-y

[23] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)*, 2017. `https://topology-tool-kit.github.io/`.

[24] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.

[25] Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen. Line graph or scatter plot? automatic selection of methods for visualizing trends in time series. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1141–1154, Feb 2018. doi: 10.1109/TVCG.2017.2653106

[26] Y. Wang, Z. Wang, L. Zhu, J. Zhang, C. Fu, Z. Cheng, C. Tu, and B. Chen. Is there a robust technique for selecting aspect ratios in line charts? *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3096–3110, Dec 2018. doi: 10.1109/TVCG.2017.2787113

[27] A. H. Young, K. R. Knapp, A. Inamdar, W. Hankins, and W. B. Rossow. The international satellite cloud climatology project h-series climate data record product. *Earth System Science Data*, 10(1):583–593, 2018. doi: 10.5194/essd-10-583-2018

[28] J. M. Zacks and B. Tversky. Bars and lines: a study of graphic communication. *Memory & cognition*, 27 6:1073–9, 1999.