

Perceptual Interpretation of Ink Annotations on Line Charts

Nicholas Kong
University of California, Berkeley
514 Soda Hall
Berkeley, CA 94720-1776
nkong@cs.berkeley.edu

Maneesh Agrawala
University of California, Berkeley
635 Soda Hall, Mail Code #1776
Berkeley, CA 94720-1776
maneesh@cs.berkeley.edu

ABSTRACT

Asynchronous collaborators often use freeform ink annotations to point to visually salient perceptual features of line charts such as peaks or humps, valleys, rising slopes and declining slopes. We present a set of techniques for interpreting such annotations to algorithmically identify the corresponding perceptual parts. Our approach is to first apply a parts-based segmentation algorithm that identifies the visually salient perceptual parts in the chart. Our system then analyzes the freeform annotations to infer the corresponding peaks, valleys or sloping segments. Once the system has identified the perceptual parts it can highlight them to draw further attention and reduce ambiguity of interpretation in asynchronous collaborative discussions.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces

General terms: Design, Human Factors

Keywords: Visualization, freeform ink annotation, visual perception, line charts, pointing, deixis, selection

INTRODUCTION

In order to communicate effectively about a visualization, it is essential to refer to marks or groups of marks representing the data. In face-to-face communication people often use deixis, or a combination of hand gestures and speech, to point to the dots, curve segments or other marks of interest, and thereby establish common ground for the discussion [5, 9]. For example, in his well known TED talks, Hans Rosling uses exaggerated hand gestures to focus his audience's attention on the marks representing various countries to show how life expectancy in those countries has changed over time¹.

Deictic pointing requires more effort in an asynchronous setting such as a web-based forum for discussing interactive visualizations [8, 14]. Although text is the primary form of communication in these systems, some of them also allow graphical annotations such as freeform ink, arrows and circles. The annotations serve as visual analogs to physical gestures as they deictically highlight the relevant marks in the charts. For instance, an author might sketch an arrow point-

¹http://www.ted.com/speakers/hans_rosling.html

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'09, October 4–7, 2009, Victoria, British Columbia, Canada.
Copyright 2009 ACM 978-1-60558-745-5/09/10 ...\$5.00.

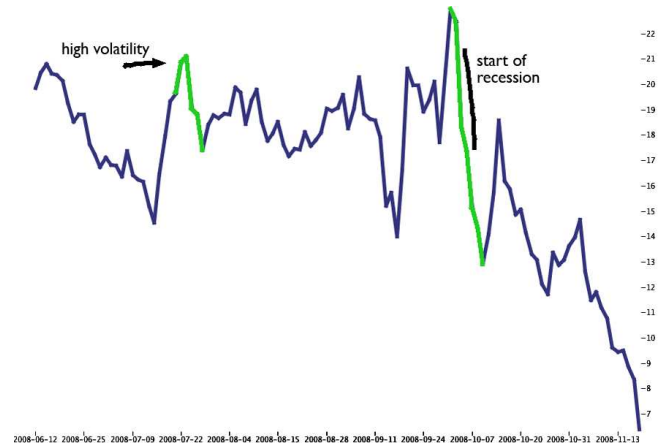


Figure 1: Citigroup stock chart from June to Nov 2008. The user sketched two annotations (in black). A human viewer easily interprets the arrow annotation (left) as pointing to a peak and the line annotation (right) as referring to a declining slope. Our algorithm considers perceptual parts in the chart to infer these interpretations (in green).

ing to a peak in a stock chart and write, “high volatility” and another stroke roughly parallel to a sharp decline and note, “start of recession” (Figure 1). Research on human visual perception suggests that people mentally decompose complex shapes, such as stock charts, into a set of *perceptual parts* such as peaks or humps, valleys, rising slopes and decreasing slopes [11]. Viewers identify the perceptual parts of the chart (e.g. the peak or the downward slope) corresponding to each annotation based on Gestalt laws of visual perception, including figure/ground articulation, proximity, similarity, continuity and closure [17].

Current interactive visualization systems are not designed to interpret ink annotations in this manner. Although many of these systems allow users to select marks via clicking, lassoing, or brushing them, such direct selection usually requires tediously precise pointing and clicking. More importantly, the mouse or pen interactions required to select marks are usually very different from the freeform ink strokes users commonly create to annotate visualization. Because these systems cannot interpret ink annotations they simply overlay the ink on the visualization for viewers to interpret.

In this paper, we present a set of techniques for algorithmically interpreting freeform ink annotations to identify the corresponding perceptual parts. Our approach is to first apply a parts-based segmentation algorithm that divides a line chart into visually salient perceptual parts such as peaks, valleys and slope segments. We then analyze the freeform annotations to determine the desired perceptual parts in the chart.

Identifying the corresponding parts of a line chart in this manner provides several benefits. The system can visually highlight the parts to further draw attention to the relevant portions of the visualization. Such highlighting is especially important in asynchronous collaboration settings as it reduces ambiguity of interpretation. Participants can check that they have interpreted the annotations the same way and are referring to the same portion of the chart. Users can annotate the chart in a natural manner without having to learn specialized selection gestures or commands. Moreover, our approach could be used to interpret existing ink annotations such as those created in *sense.us* [8]. Once parts have been selected the system could highlight the corresponding data items in additional charts showing other dimensions of the data [1, 4, 12]. It could also expand the selection to include additional nearby parts (e.g. peaks or valleys near the original selection) as a form of selection expansion [7]. Alternatively, the selected parts could serve as anchors for reflowing the ink annotations as users resize the chart [2].

RELATED WORK

Selecting marks or data items is a common task in interactive visualization systems and many of these systems provide direct manipulation controls for selecting desired elements. Dynamic queries, for example, are posed using direct manipulation widgets such as radio buttons or range sliders, with which users interactively select items based on their data values or attributes [1]. Brushing allows users to directly click, lasso or paint the marks of interest to select them [4]. These methods force users to directly specify the data values or marks of interest, whereas our approach is designed to interpret the graphical annotations placed on a visualization.

QuerySketch [16] and TimeSearcher [10] are two notable techniques for selecting time-series charts of a user-specified shape. QuerySketch allows users to sketch a query stroke and applies pattern-matching techniques to find the best-matching time-series. TimeSearcher selects all time-series that pass through a set of user-specified rectangular time-boxes. Both of these techniques are primarily designed to filter a large set of time-series charts down to a desired subset. In contrast, our approach is designed to select visually salient perceptual parts within a single chart.

Several recent techniques have focused on selecting groups of distinct visual objects based on Gestalt principles of visual perception. Watanabe et al. [15] use clustering algorithms to group together objects based on proximity and provide a variety of controls for further merging and splitting groups. Dehmeshki and Steurzlinger [6] allow users to select a few objects of interest and then expand the selection to include nearby objects based on the Gestalt principles of proximity and good continuity. However, these methods are primarily designed to work with distinct marks and cannot be used to select perceptual parts within a chart.

Our work is inspired by Saund et al.’s [13] perceptual tools for editing sketches. They introduce a lattice structure for maintaining multiple interpretations of the strokes in sketched diagrams and they provide gesture-based techniques for accessing the interpretations. For example, their path-based selection tool enables the user to draw selection paths. The system finds the shape or the sketched set of stroke frag-

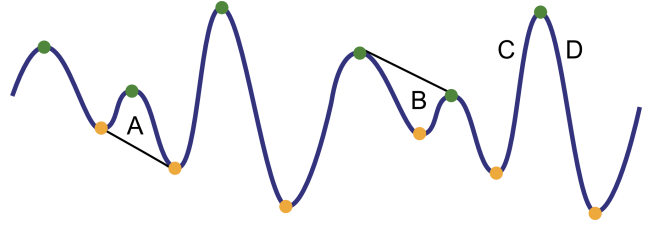


Figure 2: Line chart (blue) marked with local maxima (green) and minima (orange) of curvature. Hoffman and Singh [11] have shown that viewers mentally segment the curve into peaks (A), valleys (B) and slope segments (C, D) by mentally connecting these curvature extrema.

ments that best matches the path. Baudel’s [3] work on freeform curve editing similarly allows users to reparameterize curves by sketching changes near the curve. Neither of these techniques directly consider visually salient perceptual parts when interpreting strokes.

APPROACH

Our approach to interpreting freeform ink annotations on line charts consists of two phases: a preprocessing phase in which we segment the chart to identify perceptual parts and an interpretation phase in which we classify the annotation and then determine whether the user is pointing to a peak, valley or slope segment. We also support a simple interface for expanding the initial selection based on the structure of the visually salient perceptual parts.

Preprocess: Identifying Visually Salient Perceptual Parts

Human vision researchers Hoffman and Singh [11] have experimentally shown that viewers mentally segment shapes into parts, including peaks, valleys and slope segments, based on the local maxima and minima of curvature (Figure 2). Viewers construct peaks by mentally connecting the curvature minima on either side of a curvature maximum. Valleys are equivalent to inverted peaks (reversing figure/ground) and are formed by connecting curvature maxima on either side of a curvature minimum. Slope segments are formed by connecting adjacent curvature minima to curvature maxima (a rising slope in a line chart) or vice versa (declining slope).

Based on this analysis our preprocessing phase identifies local maxima and minima of curvature along the input chart. To analytically compute curvature we first fit a natural cubic spline to the data point of the input chart and then compute curvature $\kappa(x)$ as

$$\kappa(x) = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}} \quad (1)$$

where $y(x)$ is our spline, and y' and y'' indicate the first and second derivatives respectively.

Note that performing this analysis directly on the local extrema of the curve rather than its curvature extrema will not always produce a correct segmentation into perceptual parts. Consider the curve $y = x^3$, which has two salient parts on either side of the origin. Since the curve is monotonic no local extrema exist near the origin, but curvature extrema do exist near the origin.

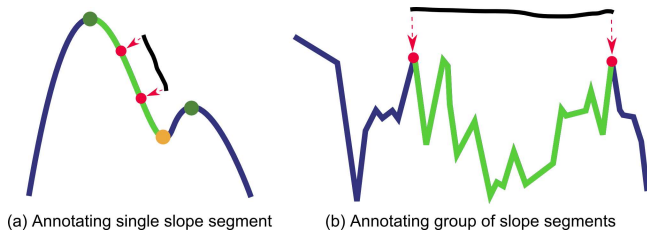


Figure 3: Parallel annotations. Both stroke endpoints lie near the chart. We project the stroke endpoints onto the chart (red points) and walk outwards to the nearest maxima (green points) or minima (orange points) to select slope segments. (left) A short stroke selects an entire slope segment. (right) A longer parallel stroke selects multiple consecutive segments between the projected endpoints.

Interpretation: Recovering Pointing Intent of Annotation

Our system allows users to draw freeform graphical annotations on a line chart visualization. Users can optionally hold the shift key while drawing a stroke to place an arrowhead at the stroke endpoint. In general, only a subset of the freeform strokes are designed to point to perceptual parts of the line chart. Thus, our system only interprets strokes that are drawn so that at least one endpoint of the stroke lies within a small threshold distance of the chart.

The primary challenge in accurately interpreting a freeform graphical annotation is to identify whether the annotation points to a peak, valley or slope segment. Our system distinguishes between two types of annotations: *parallel annotations* are strokes drawn roughly parallel to the curve and are commonly used to refer to a slope segment, while *perpendicular annotations* are strokes drawn roughly orthogonal to the curve and use often used to point to an entire peak or valley. In some cases a perpendicular annotation can also refer to a slope segment. Figure 1 contains both parallel and perpendicular annotations. Our system first classifies the type of each stroke, parallel or perpendicular, and then determines the perceptual parts of the line chart the stroke refers to.

Classifying parallel versus perpendicular annotations. We use a simple heuristic to distinguish between parallel and perpendicular annotations. We expect that both endpoints of a stroke drawn parallel to the line chart will lie near the chart. For a perpendicular stroke only one endpoint of the stroke will lie near the chart. Thus, we compute the Euclidean distance between each endpoint of the stroke and the line chart. If both endpoints are within a threshold distance we classify the stroke as a parallel annotation. If only one endpoint lies within the threshold we mark it as perpendicular. We use a threshold distance of 20 pixels (Figures 3 and 4).

Interpreting parallel strokes. Parallel strokes usually point to one or more consecutive slope segments of a chart. To identify the corresponding segments for a given stroke, we first project the two endpoints of the stroke onto the closest points on the chart. We then walk from the projected endpoints outwards in both direction to the nearest local maximum or minimum of curvature. Thus, users can point to an entire slope segment of the chart with a short stroke running along it (Figure 3 left). Users can also select multiple consecutive slope segments by drawing longer strokes. Moreover, because we only consider the projected endpoints of

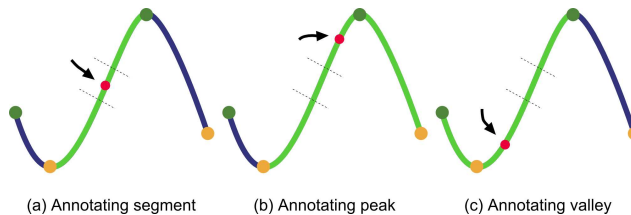


Figure 4: Perpendicular annotations. Only one stroke endpoint lies near the chart. The stroke selects a peak (left), slope segment (middle) or valley (right) depending on whether the closest stroke endpoint (red point) lies in the region of the slope segment containing the curvature maximum, segment midpoint, or curvature minimum.

the stroke and do not consider any points within the stroke, users can draw longer strokes in which only the endpoints are near the chart, to select multiple consecutive slope segments (Figure 3 right). Users may also select plateaux with this mechanism.

Interpreting perpendicular strokes. Perpendicular strokes can be ambiguous as they usually refer to a peak or a valley, but may also refer to a slope segment (Figure 4). Our approach is to assume that perpendicular strokes drawn closer to the top of a peak are more likely to refer to the entire peak. Similarly, strokes drawn closer to the nadir of a valley are more likely to refer to the entire valley. Finally, a perpendicular stroke drawn closer to the midpoint between consecutive curvature extrema is likely to refer to the slope segment.

To distinguish between these three cases, we project the closest endpoint of the perpendicular stroke onto the chart and walk outwards from the projected point in both directions to the closest curvature extremum. We compute the center of this slope segment and divide it into three regions (regions are bounded by gray dotted lines and curvature extrema in Figure 4). The small middle region covers 20% of the total distance between the minimum and maximum endpoints of the slope segment. If the projected point lies within the middle region we treat the stroke as selecting just the slope segment. If the projected point lies in the region containing the local curvature minimum we treat it as selecting the entire valley. Similarly, if it lies in the region containing the local curvature maximum we treat it as selecting the entire peak.

Selection Expansion

Our system also allows users to expand their initial selection based on the segmentation of the chart into perceptual parts. After drawing the stroke the user can press the left or right arrow keys to expand the selection outwards in the respective direction. If the annotation initially selected a slope segment, our system expands the selection by adding the next consecutive slope segment in the chart. If the annotation selected an entire peak or valley, our system expands the selection by adding the next consecutive peak or valley in the chart.

RESULTS

Figures 1, 5 and 6 show examples of line charts annotated using our system. We asked a few researchers in our lab to informally use our annotation system to create annotations on these charts. The users created the black annotation strokes and the system highlighted its interpretation of the corresponding perceptual parts in the chart (as shown in green).

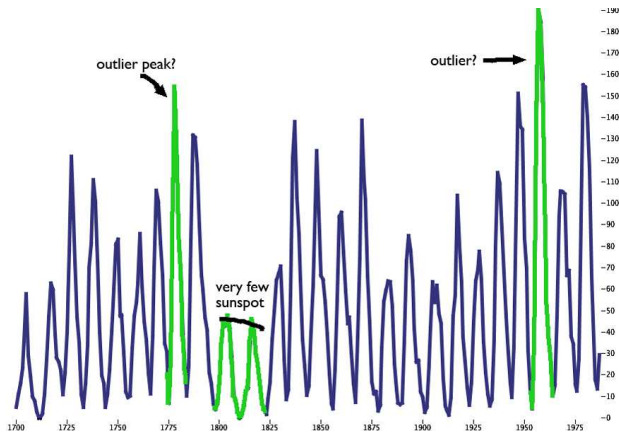


Figure 5: Annual number of sunspots from 1700 to 1987.

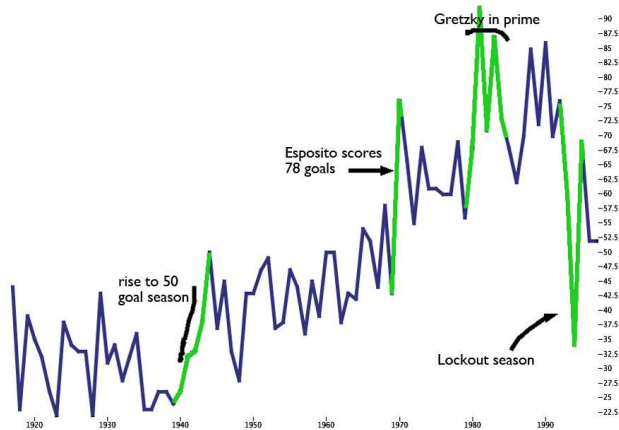


Figure 6: Annual number of goals scored by leading scorer in the National Hockey League from 1917 to 1997.

They could then choose to add a text annotations describing the highlighted portion of the curve.

These results indicate that our users were able to create annotations intended to refer to peaks, valleys and slope segments, as shown in the examples. While these results and our own experiences give us confidence that the system is generating good interpretations, we believe a more detailed study is required to draw stronger conclusions about the effectiveness of our approach. Our current implementation is intended as a proof of concept of the annotation interpretation capabilities and does not support brushing, linked highlighting or annotation reflow.

CONCLUSIONS AND FUTURE WORK

Pointing and reference are essential to effective communication. The work presented in this paper is a first step towards enabling interactive visualization systems to better understand the meaning of deictical graphical annotations. Once the system has interpreted the annotations it can further highlight the relevant portions of the chart to help asynchronous collaborators better focus on these regions. Understanding the annotation as a pointing or selection gesture is also the first step towards allowing linked highlighting, an essential technique for exploring multi-dimensional data set.

There are several directions for future work. Other types of visualizations such as scatterplots, bar graphs, stacked

area charts, etc., could benefit from similar interpretations of freeform annotations based on human visual perception of the display. We believe it would also be useful to study how annotation interpretation and highlighting of the selected chart regions might change the way people make annotations. Such annotations with highlighting have the potential to reduce ambiguity of interpretation of pointing annotations amongst asynchronous collaborators. We would like to study how well such ambiguity is reduced. Finally we would like to use the interpretations to support brushing, linked highlighting [1, 4, 12] and annotation reflow [2].

Acknowledgments: This research grew out of discussions with Doug DeCarlo and Jeff Heer. This work was supported by an NSERC PGS M and NSF grant CCF-0643552.

REFERENCES

1. C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proc. of CHI*, pages 313–317, 1994.
2. D. Barger and T. Moscovich. Reflowing digital ink annotations. In *Proc. of CHI*, pages 385–393, 2003.
3. T. Baudel. A mark-based interaction paradigm for free-hand drawing. In *Proc. of UIST*, pages 185–192, 1994.
4. R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
5. H. H. Clark and S. E. Brennan. *Grounding in communication*, chapter 7. Perspectives on socially shared cognition. APA Books, 1991.
6. H. Dehmeshki and W. Stuerzlinger. Intelligent Mouse-Based Object Group Selection. In *Proc. of Smart Graphics*, pages 33–44. Springer, 2008.
7. J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *Proc. of CHI*, pages 959–968, 2008.
8. J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. In *Proc. of CHI*, pages 1029–1038, 2007.
9. W. C. Hill and J. D. Hollan. Deixis and the future of visualization excellence. In *Proc. of IEEE Visualization*, pages 314–320, 431, 1991.
10. H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.
11. D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63:29–78, 1997.
12. A.R. Martin and M.O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proc. of IEEE Visualization*, page 271, 1995.
13. E. Saund and T.P. Moran. A perceptually-supported sketch editor. In *Proc. of UIST*, pages 175–184, 1994.
14. F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: A site for visualization at internet scale. *IEEE Trans. on Vis. and Comp. Graphics*, 13(6):1121–1128, 2007.
15. N. Watanabe, M. Washida, and T. Igarashi. Bubble clusters: An interface for manipulating spatial aggregation of graphical objects. In *Proc. of UIST*, pages 173–182, 2007.
16. M. Wattenberg. Sketching a graph to query a time-series database. In *CHI Extended abstracts*, pages 381–382, 2001.
17. M. Wertheimer. Laws of organization in perceptual forms. *A source book of Gestalt psychology*, pages 71–88, 1938.