Maximizing Resolvable Items: A Mantra for Mobile Data Visualization

Michail Schwab
Northeastern University
Boston, MA 02115, USA
michaschwab@ccs.neu.edu

Aditeya Pandey
Northeastern University
Boston, MA 02115, USA
pandey.ad@husky.neu.edu

Michelle A. Borkin
Northeastern University
Boston, MA 02115, USA
m.borkin@northeastern.edu

MobileVis '18 Workshop at CHI 2018, April 21, 2018, Montreal, QC, Canada.
https://mobilevis.github.io/

Abstract
Mobile data visualization is becoming more commonplace, but few guidelines for their design exist. In this paper, we describe the design process of taking a pan and zoom-based linear timeline from a desktop visualization to a mobile platform by turning it into a static elliptical timeline with a draggable handle for selection. The lessons we learned are generalizable for mobile data visualization and include: 1. Interaction should be simple, directly manipulate one object, and avoid two-finger gestures. 2. Linked and coordinated views are challenging, but beneficial if context is maintained. 3. Overview first, details later. 4. Because of display size and the “fat finger problem”, the number of items that can be reached with one interaction is low by default, and needs to be carefully considered and improved for effective navigation. In this paper we provide a framework to aid future design processes for mobile visualization.

Author Keywords
mobile; visualization; timeline; resolution; brushing and linking; coordinated views.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces

Introduction

Small mobile devices such as smart phones are a challenging platform for data visualization. Although many differences exist between mobile and desktop [5], two of the main reasons for this difficulty are screen size and pointing precision, in part due to occlusion [8, 11, 6, 4, 5]. On the other hand, because touch interactions can occur directly on top of the visualization, these manipulations have been described as more “intuitive” and “natural” [8, 10].

However with the expectation of “natural” and direct interactions come responsibilities for visualization creators. For example, clicking through menus is less accepted on mobile than on desktop. Complex interactions, such as pan and zoom, are to be used with care [5].

How do visualization creators deal with display size and pointing precision restrictions on mobile, while enabling “natural” and direct interactions?

One branch of related work aims to reduce the need for precise interaction through effective summarization techniques and proposes more use of the “overview first, details later” mantra on mobile [3]. Another branch targets the interaction precision with clever device designs, such as touch interactions occurring on the back of a device rather than on the front screen in order to prevent occlusion [11, 2]. Although some related work provide a good overview of the problems in mobile visualization [5], there are few with explicit solutions to these challenges.

Existing popular mobile photo applications created by Google and Apple enable fast navigation of many photos [7, 1], but do not provide much overview or visual guidance that may help navigation with partial knowledge. The trade-offs between simple but efficient navigation and visual guidance are an important topic to consider, but outside of the scope of this work. In this paper, we assume there to be enough benefits to using visualization for photo summary to warrant an exploration of this space. We describe a case study of timeline design for desktop and mobile, and summarize our key design process findings. Specifically we contribute:

- The step-by-step design process to transform a horizontal, zoom- and brush-based timeline of a desktop photo application to an elliptical handle-based timeline, and explain what problems it solves,
- The “Maximizing Resolvable Items” Mantra with an explanation of its metric for selection quality,
- Other lessons learned and design recommendations, including simple and direct interactions, prioritized and context-preserving linked and coordinated views, and overview + detail.

Design Process of a Mobile Timeline

As part of a project to develop a personal photo application with timeline visualization component, we needed to migrate our application from desktop to mobile. During this transition we quickly realized the need of significant adjustments to the timeline visualization. The timeline was part of a linked views design which took advantage of a map to encode geospatial information, a timeline to encode time, and a table view to show photos. We show the design process and reasoning at each step in Table 1. After Step 2, we conducted seven semi-structured user interviews of 45-60 minutes. Results are mentioned when relevant.

Step 0: Original Design. On the computer, the timeline had complex pan and zoom interactions, and the *selection* of photos was indirect through brush interactions.

Step 1: Merge Brush & Zoom. After moving to mobile, we realized maintaining both zoom and brush interactions is too complex. To simplify, we merged these two features by hiding the brush and assuming the brush selection to be the currently visible items. This was successful and in the study, participants had no trouble understanding how to select via zooming. Still, pan and zoom make for high *interaction complexity*, and *selection* is indirect via zoom.

Additional Handle-Based Interaction Features

In addition to the functionality described in the main text, we highlight two additional direct manipulation features below:

Inside vs. Outside: Users can either select one of the “trip” clusters or select a month only based on time, depending on whether their finger is on the inside or outside of the ellipse. The handle then snaps either to the inside of the ellipse, which contains the trip clusters, or to the outside of the ellipse, which contains the month segments. This allows selection either by by time and location, or by time alone.

Select All and Handle Size: On start-up, the handle sits on the "Select All" position in Figure 1. During the drag interaction, it then turns into a small circle as it moves along the ellipse to give precise visual feedback. Releasing the handle rescales it to its full size at the current position for future dragging interactions. A full clockwise rotation switches to the next year, counterclockwise to the previous year.

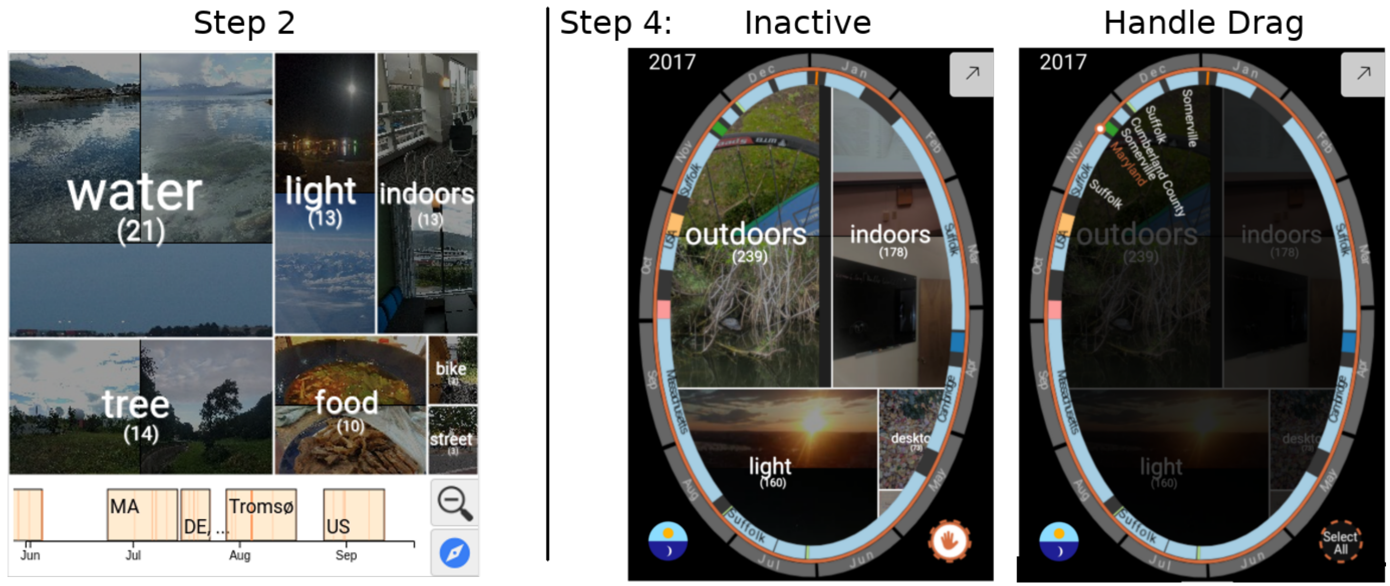


Figure 1: The full mobile application design around Step 2 (left), and the design after Step 4 (right), inactive and while dragging the handle. Dragging shows the location labels for trips close to the handle position, and the handle snaps to the nearest cluster and highlights their label.

Additionally, on the small screen and due to the “fat finger problem” [11], few items were reachable without interaction. Users had to do a disproportionate and unpleasant amount of panning and zooming for meaningful selections (see [5]).

Step 2: Cluster Items. We chose to cluster photos in our application by location for two reasons: first, the mobile platform did not support displaying the timeline, map, and photos on the same screen due to screen size constraints, and we wanted to provide as much geo-spatial information as possible without the constant need of a map. Second, we wanted to make it simpler to make meaningful selections, and had learned through user interviews that trips to different locations are a common selection. Thus clustering reduced the number of complex zooming interactions, although it did not eliminate it. The main benefit in this step is that the selection interaction is *direct* via click, as opposed to having to select a range via zooming.

Step 3: Layout Change. We observed at this step that the timeline was unable to resolve many items and needed to dedicate more screen space. We decided a vertical design for the portrait layout of phones was insufficient. We considered a rectangular layout around the edges of the screen, but found the interaction to be more intuitive with a circular layout. We also had good use for the corners a circle would create, such as filtering via day time, and toggles for hiding the timeline to focus on photos once a time selection is made, see Figure 1. We decided to use an elliptical layout.

Resolved Items below are an order of magnitude estimate.

Assuming a timeline width of 1000 pixels and 5 pixels precision, the resolution is **200** items on desktop.

With a timeline width of about 400 virtual pixels and a precision of about 20 pixels due to the “fat finger problem”, the resolution is only about **20** items on mobile.

Due to the increased length of the timeline as ellipse, the number of items we were able to resolve more than tripled and went up to around **65**.

Assuming the precision goes up to the level of mouse-based pointing because of the visual feedback of the handle, the number of resolved items would go up to **260**, a level comparable with that of the linear timeline on desktop.

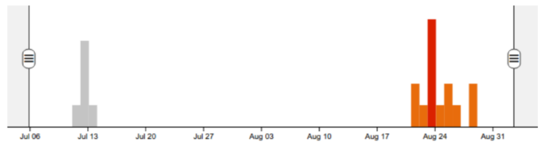
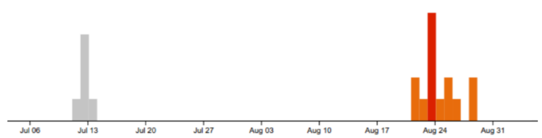
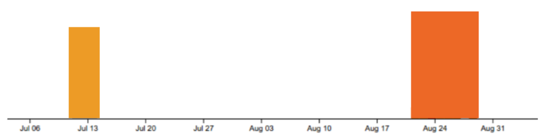
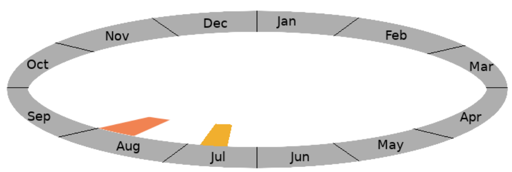
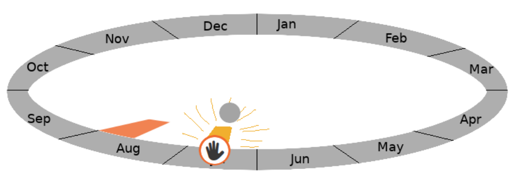
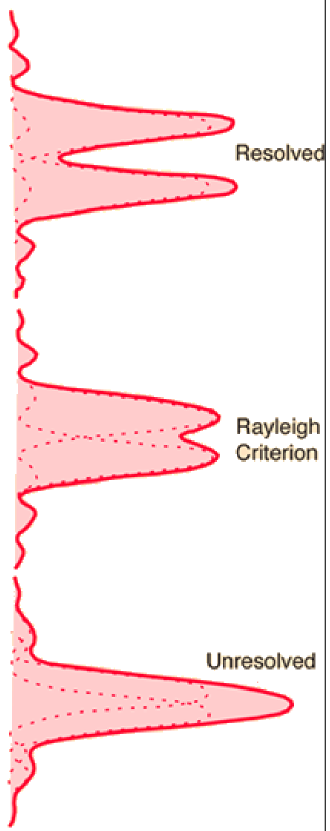
Step	Timeline Illustration	Change	Goal	Resolved Items	Interaction Complexity	Selection
0		Original Design	*N/A*	**200**	High	Indirect
1		Merge Brush & Zoom	Simpler	**20**	High	Indirect
2		Cluster Items	Simpler	**20**	High	Direct
3		Layout Change	More Space	**65**	High	Direct
4		Zoom & Click → Handle	More Precision	**260**	Low	Direct

Table 1: The progression of the timeline design from linear with pan and zoom and brush on desktop (Step 0) to aggregated, elliptical and handle-based on mobile (Step 4). Through the design process, the interactions become simpler and more direct, and the number of items that are reachable with one interaction goes back up to the number it was on desktop.

Resolution in Physics
In Physics, the *resolution* is the minimum distance needed between points to distinguish them. Below, you see two light sources as they move closer together [9]:



The precision of a pointing device is analogous to the precision in physics, in that a precise device allows pointing to two items that might not be differentiable with a less precise device.

Step 4: Zoom & Click → Handle. In the final step, we redesigned the interactions of the system. Instead of using zoom, we decided to place a handle on the timeline, which the user could drag along the ellipse, and which could snap either to months or to trips. While dragging the handle, the user is able to move their finger on the inside of the ellipse, and the handle position is updated according to the angle of the finger position relative to the center. This allows specifying unique positions along the ellipse without obscuring the selection with the finger by moving it in a circle closer to the center of the ellipse, rather than moving directly over the target. Furthermore, by updating the current handle position, and through snapping and highlighting of the nearest item, the visualization is able to provide valuable visual feedback during the selection process. Through visual feedback and unobscured vision, we greatly improved the selection precision.

The comparison of the actual implementations between Step 2 and 4 is shown in Figure 1. In the original design, we used a regular linked view layout, whereas the elliptical design is more integrated and the content is embedded in the ellipse. When the timeline is either manually or automatically deemed not needed, it makes room for the content to fill the entire screen by increasing the ellipse radius until it goes off screen and is no longer visible.

Maximizing Resolvable Items: A Mantra for Mobile Data Visualization

In our case study, after moving the timeline to the mobile platform it was difficult to navigate to items on the timeline due to the low precision of item selection and limited screen space. The smaller screen required many pan and zoom interactions for effective selections. We found that a general and instructional way to think about this problem is the *resolution*, similar to resolution in Physics (see sidebar):

How many items N can we resolve, i.e., reliably navigate to with pointing, on the visualization? This primarily depends on two factors: the total amount of space A used by the visualization, and the precision D of the input:

$$N_{\text{resolved}} = \frac{A_{\text{space}}}{D_{\text{precision}}} \qquad (1)$$

For example, on a one-dimensional visualization with 1000 pixels available, an input device with a precision of about 10 pixels leads to about 100 items being selectable. To double the number of resolved items, we could either double the length of the visualization to 2000 pixels, or we could double the precision to 5 pixels.

This concept is applicable to mobile data visualization design as both screen size as well as precision of the pointing device are much lower than on desktop, severely limiting the number of resolvable items. According to information theory, this means that each interaction carries less information, the consequence being that each interaction distinguishes between fewer items. In practice this means that users can either only click on large aggregations of data, or that complex interactions are needed for navigation, such as pan and zoom, which are often not desired [5].

Other Lessons Learned

Through the process and learning experience of redesigning this timeline, and our photo application overall, for a mobile device we learned many valuable lessons. We summarize these lessons learned and design insights in the following section as a useful reference guide for other developers of visualizations on mobile devices.

Direct Manipulation: Interaction through finger enables a more direct manipulation of the content. In user testing, we observed that this has significant consequences.

In our experience, touch interactions lead to a much higher expectation of interactions. While clicking through menus is acceptable with indirect pointing devices, the desired manipulation of the object with the direct input is expected to feel much more natural.

Due to these different expectations, we believe there is a large space of unexplored interactions that could be very well received as “natural” on the mobile platform and positively impact adoption. For example, would Tinder’s “swipe” gesture have been successful on a desktop application? What about Apple iPod’s rotation selection?

In summary, we learned that **interactions should be direct manipulations of a single object whenever possible, they should be simple, and avoid two-finger interactions**, in line with [5]. Two finger interactions generally mean that two hands have to be used to operate the phone, because one is needed to hold it. In contrast single finger operations can be used on the same hand that is holding the phone, leading to increased comfort and efficiency.

Linked and Coordinated Views: The use of linked views is common when ample screen space is available, but is challenging to design and implement for small screens. Several strategies can help based on our experiences:

1. Prioritize: Which are the crucial views? Which view’s information can be partially communicated through other views? Through interviews, we found that the time a photo was taken is the most important and common information users remember about their photos. As a result our timeline was made a persistent and primary component of the visualization.

2. Coordinate instead of Link: It might be advantageous to provide multiple views because, depending on context, the views may enable more efficient navigation. However, it may not be feasible to display all views at once. In these cases, we have found coordinated views to be advantageous for filtering data across different visualizations. With coordinated views, it is important to maintain context, e.g., by visually reminding the user of current filters.

Overview + Detail: Selection should be as easy as possible. Even with sufficient space available to enable the selection of 200 individual items, it may be more appropriate to enable selection of a few clustered or binned groups of the items [3]. The goals and tasks of the visualization need to be carefully considered, and aggregation should be done as much as possible while still supporting the tasks.

Discussion and Conclusion

To address the limited screen space, more imprecise interactions, and expectation of more direct interactions, a good mobile data visualization should:

1. Minimize complex interactions and instead choose simple and direct ones,
2. Minimize the number of items needed for navigation via aggregation using overview + detail,
3. Maximize the use of the screen space with clever designs in prioritized linked and coordinated views,
4. Maximize the precision of the input using visual feedback.

The lessons learned have proven useful in our experience. We look forward to the community conducting further formal evaluations to quantify benefit and role of these design recommendations building upon previous work on some aspects such as the benefit of overview + detail on mobile [3]. Through thoughtful reflection of these lessons and application of the Maximizing Resolvable Items mantra, we hope others will be able to effectively design interactive visualizations for mobile devices.

Acknowledgements

We thank Matthew Brehmer for fruitful discussions and design inspiration. This work was supported by the College of Computer and Information Science at Northeastern University.

REFERENCES

1. Apple. 2018. Apple Photos. (2018). https://www.apple.com/ios/photos/
2. Patrick Baudisch and Gerry Chu. 2009. Back-of-device Interaction Allows Creating Very Small Touch Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1923–1932. DOI: http://dx.doi.org/10.1145/1518701.1518995
3. Stefano Burigat and Luca Chittaro. 2013. On the Effectiveness of Overview+Detail Visualization on Mobile Devices. *Personal Ubiquitous Comput.* 17, 2 (Feb. 2013), 371–385. DOI: http://dx.doi.org/10.1007/s00779-011-0500-3
4. T. Capin, K. Pulli, and T. Akenine-MÃűller. 2008. The State of the Art in Mobile Graphics Research. *IEEE Computer Graphics and Applications* 28, 4 (July 2008), 74–84. DOI: http://dx.doi.org/10.1109/MCG.2008.83
5. L. Chittaro. 2006. Visualizing information on mobile devices. *Computer* 39, 3 (March 2006), 40–45. DOI: http://dx.doi.org/10.1109/MC.2006.109
6. Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. 2007. Direct-touch vs. Mouse Input for Tabletop Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 647–656. DOI: http://dx.doi.org/10.1145/1240624.1240726
7. Google. 2018. Google Photos. (2018). https://photos.google.com/
8. Robert Mack and Kathy Lang. 1989. A Benchmark Comparison of Mouse and Touch Interface Techniques for an Intelligent Workstation Windowing Environment. *Proceedings of the Human Factors Society Annual Meeting* 33, 5 (1989), 325–329. DOI: http://dx.doi.org/10.1177/154193128903300520
9. Carl Rod Nave. 2000. HyperPhysics. (2000).
10. Dominik Schmidt, Florian Block, and Hans Gellersen. 2009. A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces. In *Human-Computer Interaction – INTERACT 2009*, Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, Raquel Oliveira Prates, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 582–594.
11. Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007. Lucid Touch: A See-through Mobile Device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 269–278. DOI: http://dx.doi.org/10.1145/1294211.1294259