Demonstrating VisTiles: Visual Data Exploration Using Mobile Devices

Ricardo Langner, Tom Horak, Raimund Dachselt
Interactive Media Lab, Technische Universität Dresden, Germany
langner@acm.org, horakt@acm.org, dachselt@acm.org

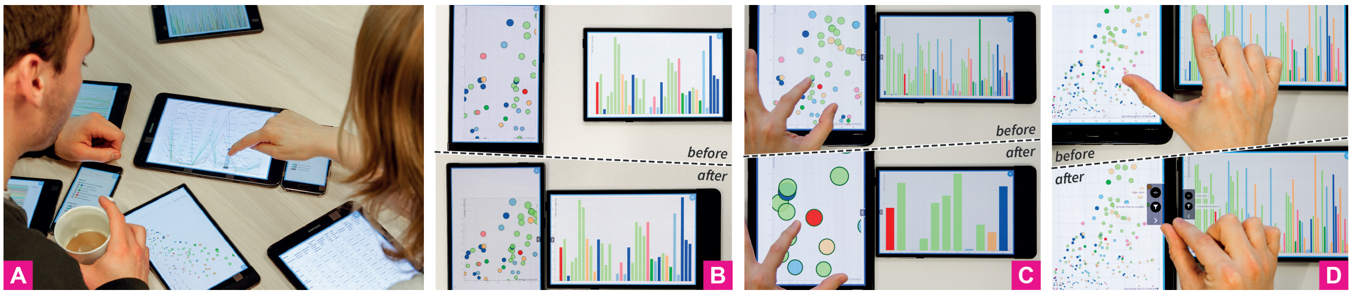

Figure 1: The VisTiles framework: (a) Allowing multiple users to visually analyze data with mobile devices; (b) Aligning visualizations on side-by-side devices; (c) Filtering data based on a visualization viewport (*Filter-by-viewport*), offscreen scatterplot items are filtered in the bar chart; (d) Initializing device combination with a *pinch gesture* on two apposed devices.

ABSTRACT

We demonstrate the prototype of the conceptual VisTiles framework. VisTiles allows exploring multivariate data sets by using multiple coordinated views that are distributed across a set of mobile devices. This setup allows users to benefit from dynamic and user-defined interface arrangements and to easily initiate co-located data exploration sessions. The current web-based prototype runs on commodity devices and is able to determine the spatial device arrangement by either a cross-device pinch gesture or an external tracking system. Multiple data sets are provided that can be explored by different visualizations (e.g., scatterplots, parallel coordinate plots, stream graphs). With this demonstration, we showcase the general concepts of VisTiles and discuss ideas for enhancements as well the potential for application cases beyond data analysis.

1 INTRODUCTION

Typical visualization workplaces involve single users, sitting on classic desktop computers, often working with multiple visualization views that are synchronized (a concept called multiple coordinated views, MCV). Such desktop setups, however, are not ideal for situations where multiple users need to work together. This raises the questions of: *How can we allow multiple users to collaboratively explore data visualizations?* and *What device setups can be used?* A possible answer to these questions is the recently introduced VisTiles framework [5], which is part of a series of approaches focusing on the combination of multiple mobile devices (e.g., [2, 6–9, 12–14, 17, 18]). VisTiles supports co-located collaborative data exploration by distributing components of a MCV interface across mobile devices. The proposed interaction concepts make use of a physical workspace that allows combining and adapting visualizations based on the spatial arrangement of devices.

In this demonstration, we showcase the prototype implementation of the VisTiles framework for the first time. The goal is to present both concepts and techniques as well as ideas for potential enhancements of the approach. For that, we extended and further improved a previous version of the prototype [5]. While the original prototype required an external device tracking to initiate view combinations, the current version[1] allows an alternative mechanism. Although this step involves additional user interactions, it really allows ad-hoc combinations with available commodity devices—an external tracking system is not required anymore. It also gives users more control as they explicitly confirm that devices shall be combined. Furthermore, the application allows users to choose between different data collections and visualizations at runtime.

2 VISTILES: BASIC IDEA AND CONCEPT

In general, VisTiles builds on the idea of *distributed user interfaces*, thus multiple visualizations are distributed across a number of mobile devices. Because these devices can be arranged in relation to each other, we call them **vis**ualization **tiles**. To reduce the interface complexity per tile and maximize the size of each visualization, VisTiles is designed in a way that each tile shows one component of the user interface (menu or visualization view). The result is an alternative visualization interface allowing multiple users to visually analyze data with multiple mobile devices. By addressing the concepts of “intelligent use of space” [4], “space-to-think” [1], and “tangible views” [15], VisTiles enables users to benefit from a flexible and user-defined view arrangement as well as a physical workspace allowing to pick up and spatially organize views.

Permission to make digital or hard copies of part or all 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. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
AVI '18, May 29-June 1, 2018, Castiglione della Pescaia, Italy
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5616-9/18/05.
https://doi.org/10.1145/3206505.3206583

[1]Prototype sources are freely available: https://github.com/imldresden/vistiles; Project website with further information and materials: https://imld.de/vistiles

2.1 Cross-Device Visualization Adaptations

Since visualization views of a MCV interface are typically linked (coordinated), VisTiles also synchronizes tiles. Interactions with one of the tiles also affect views of other tiles. In the simplest case, selections or highlighted data items are shared across connected devices, which enables *linked brushing*—an essential technique used for MCV. Furthermore, the action of arranging two or more tiles side by side can be interpreted as an explicit spatial device configuration to enable other visualization adaptations. The main motivation is to exploit the fact that, if different visualization views are relevant for certain user goals (e.g., visually compare data items [16]), it is beneficial to bring these views close to each other. Besides this, arranging tiles can support further visualization-specific tasks, such as identify outliers, reveal correlations and distributions, or details on demand. The following list gives a brief overview of selected visualization adaptations described in our VisTiles article [5]:

Alignment: Visualizations of apposed tiles can be *aligned* (Figure 1b) by translating or scaling them depending on technical and spatial properties (e.g., resolution, position, and orientation).

Rearrangement: In addition to alignment, the position of certain elements of visualizations (e.g., data items, axes) can be adapted, for instance, by rearranging bars of different bar charts in a way that all charts use the same order for bars. Again, this can improve readability and it supports comparison of displayed values.

Display Extension: To counteract the limited screen real estate of mobile devices, visualizations can be expanded across the screens of multiple tiles. This process should also consider technical properties of involved devices (e.g., pixel density).

UI Offloading: To maximize the size of each visualization, UI components such as can be offloaded to other tiles, e.g., controls for adjusting data mappings, encodings, or color schemes.

Overview & Detail: If applicable, synchronized tiles automatically indicate positions and sizes of other viewports by displaying corresponding bounding boxes within the visualizations (see Figure 2 workspace 2). This concept allows a remote manipulation of such views by interacting with the bounding boxes.

Filtering: The 'filter-by-viewport' mechanism dynamically changes the visibility of items based on the specific viewport of one of the tiles. E.g., by zooming and panning a scatterplot, off-screen items are immediately filtered in other synchronized views (Figure 1c).

2.2 Manage Device Synchronizations

To enable the use of a varying number of devices and provide basic multi-user support, VisTiles uses the concept of workspaces (Figure 2). The idea is that only devices within the same workspace are synchronized. By adding or removing devices to/from a workspace, users can control the synchronization between tiles.

Another interesting design aspect is the way of initializing device combinations. So far, the VisTiles concepts used an automatic detection of side-by-side combinations. With today's mobile devices this typically requires to somehow enhance the technical setup by, for example, using an external camera tracking solution [5, 13] or using device cameras to track external markers mounted on the ceiling [6]. In addition to this, we now extended the prototype and implemented a non-tracking-based technique that already powered other applications running on multiple co-located mobile devices [10, 11]. By performing a *pinch gesture* on two side-by-side

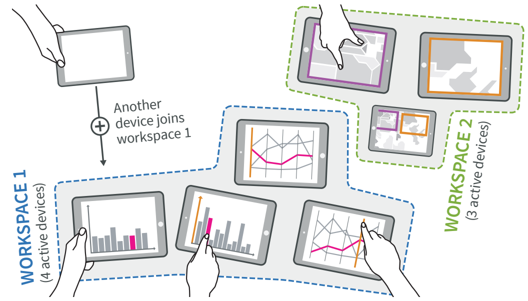

Figure 2: Workspaces control view (device) coordinations. While workspace 1 illustrates *linked brushing*, workspace 2 shows a *overview & detail* setting.

arranged devices (Figure 1d), users can explicitly combine visualization tiles. Similar to *Stitching* [3], the system detects the gesture and then calculates relative device positions. When devices are combined, an option menu at the corresponding sides of the displays (Figure 1d) allows users to activate or deactivate useful visualization adaptations.

3 PROTOTYPE IMPLEMENTATION

Similar to other cross-device research prototypes, our implementation builds on modern web technologies and is realized with a client-server architecture. Therefore, all mobile devices with up-to-date web browsers are supported. The server side is driven by a Node.js application (written in TypeScript), which handles client communication via WebSockets and covers the cross-device application logic (e.g., workspaces, linked brushing). The server also provides access to the different example data collections. On the client-side (mobile devices), browsers connect to this server and display the user interface. The client-application uses D3.js for different visualizations, which currently includes the following charts: scatterplots, line charts, bar charts, parallel coordinate plots, stream graphs, and tables (spreadsheets).

In order to support certain visualization adaptations (e.g., alignment or display extension), the application requires information regarding technical properties of connected devices. While properties such as the screen resolution can be extracted at runtime, other aspect such as bezel widths are typically not easily available. Therefore, we manually created a property catalog for devices we used during the development.

4 CONCLUSION

This demonstration presents the current VisTiles prototype, which allows to explore multivariate data collections by using a set of mobile devices. The basic idea is to distribute multiple coordinated visualization views across co-located mobile devices. The extended prototype implementation now includes different data collections and provides an alternative mechanism to initiate device combinations. Finally, with this demonstration, we want to motivate discussions on (i) the use of spatially arranged mobile devices for information visualization as introduced by VisTiles, and (ii) the utility for application cases beyond data analysis.

Acknowledgements This work was supported in part by DFG grant DA 1319/3-3 (GEMS 2.0).

REFERENCES

[1] Christopher Andrews, Alex Endert, and Chris North. 2010. Space to Think: Large High-resolution Displays for Sensemaking. In *Proc. CHI '10*. ACM, USA, 55–64. https://doi.org/10.1145/1753326.1753336

[2] Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2773–2782. https://doi.org/10.1145/2556288.2557170

[3] Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. 2004. Stitching: Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 23–31. https://doi.org/10.1145/989863.989866

[4] David Kirsh. 1995. The intelligent use of space. *Artificial Intelligence* 73, 1 (1995), 31–68. https://doi.org/10.1016/0004-3702(94)00017-U

[5] Ricardo Langner, Tom Horak, and Raimund Dachselt. 2018. VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE TVCG* 24, 1 (Jan 2018), 626–636. https://doi.org/10.1109/TVCG.2017.2744019

[6] Ming Li and Leif Kobbelt. 2012. Dynamic Tiling Display: Building an Interactive Display Surface Using Multiple Mobile Devices. In *Proc. MUM '12*. ACM, USA, Article 24, 4 pages. https://doi.org/10.1145/2406367.2406397

[7] Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2011. Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1787–1796. https://doi.org/10.1145/1978942.1979201

[8] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. 2018. SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 354:1–354:14. https://doi.org/10.1145/3173574.3173928

[9] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-device interaction via micro-mobility and f-formations. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 13–22. https://doi.org/10.1145/2380116.2380121

[10] Heidi Selmer Nielsen, Marius Pallisgaard Olsen, Mikael B. Skov, and Jesper Kjeldskov. 2014. JuxtaPinch: An Application for Collocated Multi-device Photo Sharing. In *Proc. MobileHCI '14*. ACM, USA, 417–420. https://doi.org/10.1145/2628363.2633569

[11] Takashi Ohta and Jun Tanaka. 2012. Pinch: An Interface That Relates Applications on Multiple Touch-Screen by 'Pinching' Gesture. In *ACE*. LNCS, Vol. 7624. Springer, 320–335. https://doi.org/10.1007/978-3-642-34292-9_23

[12] Tommaso Piazza, Morten Fjeld, Gonzalo Ramos, AsimEvren Yantac, and Shengdong Zhao. 2013. Holy Smartphones and Tablets, Batman!: Mobile Interaction's Dynamic Duo. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*. ACM, 63–72. https://doi.org/10.1145/2525194.2525205

[13] Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proc. ITS '14*. ACM, USA, 45–54. https://doi.org/10.1145/2669485.2669500

[14] Roman Rädle, Hans-Christian Jetter, Mario Schreiner, Zhihao Lu, Harald Reiterer, and Yvonne Rogers. 2015. Spatially-aware or Spatially-agnostic?: Elicitation and Evaluation of User-Defined Cross-Device Interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3913–3922. https://doi.org/10.1145/2702123.2702287

[15] Martin Spindler, Christian Tominski, Heidrun Schumann, and Raimund Dachselt. 2010. Tangible Views for Information Visualization. In *Proc. ITS '10*. ACM, USA, 157–166. https://doi.org/10.1145/1936652.1936684

[16] Christian Tominski, Camilla Forsell, and Jimmy Johansson. 2012. Interaction Support for Visual Comparison Inspired by Natural Behavior. *IEEE TVCG* 18, 12 (Dec 2012), 2719–2728. https://doi.org/10.1109/TVCG.2012.237

[17] Paweł Woźniak, Nitesh Goyal, PrzemysPaweław Kucharski, Lars Lischke, Sven Mayer, and Morten Fjeld. 2016. RAMPARTS: Supporting Sensemaking with Spatially-Aware Mobile Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2447–2460. https://doi.org/10.1145/2858036.2858491

[18] Paweł Woźniak, Lars Lischke, Benjamin Schmidt, Shengdong Zhao, and Morten Fjeld. 2014. Thaddeus: A Dual Device Interaction Space for Exploring Information Visualisation. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. ACM, 41–50. https://doi.org/10.1145/2639189.2639237