To appear in IEEE Transactions on Visualization and Computer Graphics

VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration

Ricardo Langner, Tom Horak, Raimund Dachselt

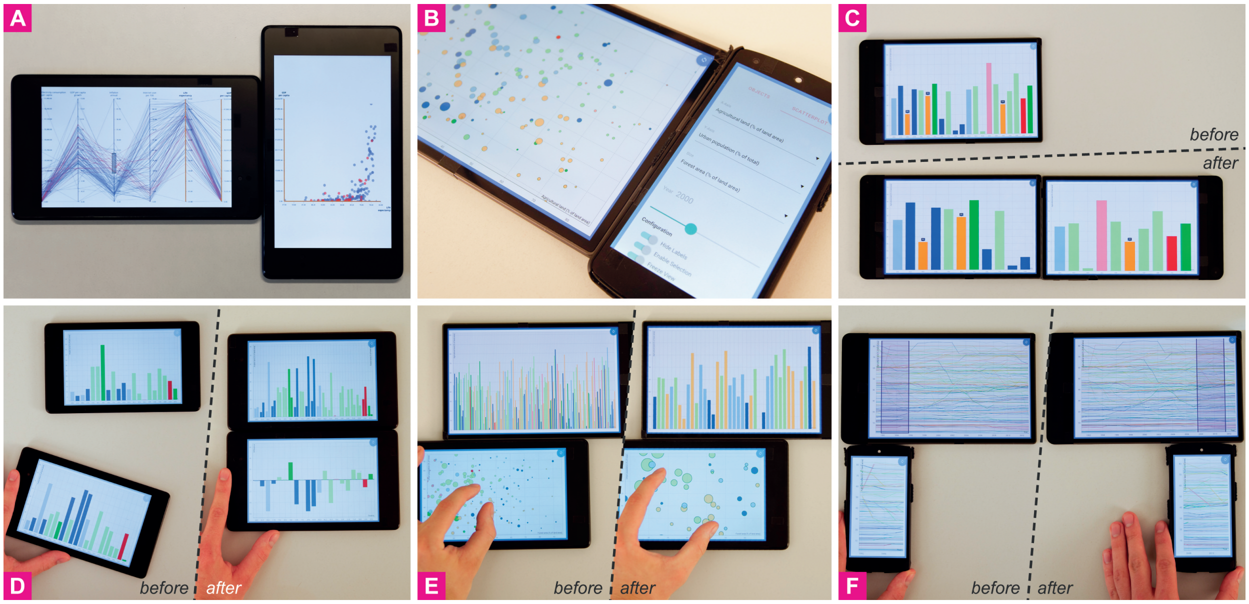

Fig. 1. VisTiles, a conceptual framework supporting visual data exploration through: (a) aligning visualizations, (b) reducing visual clutter by UI offloading, (c) extending display space, or (d) displaying computed data; device combinations can also be used to (e) filter data items (here based on the viewport) or (f) manipulate overview & detail views by spatial movements.

Abstract— We present VisTiles, a conceptual framework that uses a set of mobile devices to distribute and coordinate visualization views for the exploration of multivariate data. In contrast to desktop-based interfaces for information visualization, mobile devices offer the potential to provide a dynamic and user-defined interface supporting co-located collaborative data exploration with different individual workflows. As part of our framework, we contribute concepts that enable users to interact with coordinated & multiple views (CMV) that are distributed across several mobile devices. The major components of the framework are: (i) *dynamic and flexible layouts for CMV* focusing on the distribution of views and (ii) an interaction concept for *smart adaptations and combinations of visualizations* utilizing explicit side-by-side arrangements of devices. As a result, users can benefit from the possibility to combine devices and organize them in meaningful spatial layouts. Furthermore, we present a web-based prototype implementation as a specific instance of our concepts. This implementation provides a practical application case enabling users to explore a multivariate data collection. We also illustrate the design process including feedback from a preliminary user study, which informed the design of both the concepts and the final prototype.

Index Terms—Mobile devices, coordinated & multiple views, multi-display environment, cross-device interaction.

1 Introduction

A common user interface (UI) characteristic in the domain of visual data analysis—and information visualization (InfoVis) in general—is the frequent use of multiple visualization views at the same time. To support users, the approach of *coordinated & multiple views* (CMV) [47] synchronizes user interactions across views by using mechanisms such as linked brushing. Many visualization systems—and CMV in particular—are created for traditional workplaces, linked to the idea of desktop computers where people work in a separated and independent manner. However, today's working environments have developed further. People often work together and discuss during face-to-face working sessions and teams have flexible and sometimes varying number of group members. In addition, teamwork is supported by providing a lot of space for flexible use (i.e., open space with large tables and free walls) as well as specialized working artifacts such as white boards or even large interactive displays. This illustrates that the needs for teamwork and particularly collaborative visualization [20–22, 56] go beyond the capabilities of a classic desktop interfaces. By considering new input and output technologies, InfoVis can benefit from a more natural form of interaction (e.g., [9, 31, 46]).

At this point, we believe that mobile devices such as smartphones or tablets have a great and yet underexplored potential for InfoVis. On the one hand, continuous technical advances lead to more powerful devices with ultra-thin, bezel-less, and high-resolution displays. On the other hand, important cross-device interaction techniques have been explored, particularly techniques using spatially-aware mobile devices

- *Ricardo Langner is with the Interactive Media Lab at the Technische Universität Dresden, Germany. E-mail: ricardo.langner@tu-dresden.de.*
- *Tom Horak is with the Interactive Media Lab at the Technische Universität Dresden, Germany. E-mail: tom.horak@tu-dresden.de.*
- *Raimund Dachselt is with the Interactive Media Lab at the Technische Universität Dresden, Germany. E-mail: dachselt@acm.org.*

that incorporate the spatial relation with the surrounding (e.g., [30,44, 54,58]). Therefore, we believe that future mobile devices will facilitate an interaction style similar to the way we handle paper today for sense-making activities [15, 22]. As devices are likely to be available in large quantities, it will be possible to use and easily combine multiple devices at the same time. Moreover, mobile devices have become more lightweight, but are still graspable and thus allow us to pick them up, move them around, and more importantly, physically organize a set of devices. As emphasized by Kirsh [26], the way "how we manage the spatial arrangement of items around us, is not an afterthought; it is an integral part of the way we think, plan and behave."

In the context of InfoVis and visual data analysis in particular, we envision a new visualization interface enabling the *co-located collaborative analysis* of multivariate data. Users interact with several coordinated visualization views that are *distributed across multiple mobile devices* (Fig. 1). This is, however, not limited to one device per person. Since users bring their own devices and work environments also supply a set of devices, the number of involved devices can be adapted individually. Furthermore, by moving and placing devices on a table, users *arrange and combine* them in useful and meaningful ways, thus addressing the concept of the "intelligent use of space" [26] and "space to think" [1]. This *physical workspace* also supports individual needs of diverse forms of teamwork. Users might 'bring devices and join' or 'pick up devices and leave' working sessions at any time; or they might flexibly switch between different collaboration styles [19].

In this paper, we explore the possibilities of this envisioned, novel class of InfoVis interfaces. Our major contribution is a conceptual framework called VISTILES, in which we describe how coordinated & multiple views can be implemented using mobile devices. This framework comprises two main components:

- *Dynamic and individual layouts for CMV* The first component provides the basics of distributing visualization views across mobile devices. The resulting physical workspace allows users to spatially arrange devices and thus visualization views.
- *Smart adaptations and combinations of visualizations* The second component comprises both interaction and visualization concepts that create an extended version of CMV. Based on side-by-side arrangements and spatial movements of devices, visualization views are adapted, synchronized, or combined to ease data exploration.

While focusing on multivariate data, the concepts of our framework also address different InfoVis tasks and goals, for instance: to filter and dynamically query data [52]; reconfigure, encode, and abstract/elaborate from the interaction categories by Yi et al. [59]; overview, zoom, details-on-demand, and relate [53]; or visual comparison [19,57]. Furthermore, we also contribute a fully functional web-based prototype implementation, which demonstrates how this conceptual framework can be realized. This VISTILES prototype is also freely available on GitHub[1].

In the following sections, we first give an overview of prior work. We then describe the fundamental concepts and design of the VISTILES framework. Furthermore, we explain the design process in order to illustrate rationales for design decisions and how the concepts evolved. We report on feedback from a preliminary user study and provide details about the prototype implementation of our concepts. Finally, we discuss the limitations and capabilities of VISTILES as well as reflect on design alternatives.

2 BACKGROUND

Our work builds on various results and insights of existing research in the fields of visualization and human-computer interaction. Here we review prior work with a focus on mobile displays and (i) their use in the context of information visualization, (ii) their combination or arrangement, and (iii) the utilization of their awareness of space. We further report on research that is placed at the intersections of these three areas.

Research regarding **InfoVis on mobile displays** has evolved from using single PDAs for simple visualization techniques [5,6]. More recently, research focused on the design of multi-touch techniques for specific visualization techniques: Baur et al. [4] for stacked graphs, Drucker et al. [8] for bar charts, or Sadana and Stasko [48, 50] for scatterplots. These works minimize the use of traditional desktop UI widgets by introducing independent multi-touch interfaces that allow direct interactions on visualization elements, such as axes, canvas, or data items. As showed by Drucker et al. [8], these touch interfaces can be faster, less error-prone, and user preferred compared to a desktop interface. However, there is still no general set of multi-touch gestures for visualizations that might guide the design of new systems. In recent research, Sadana and Stasko [49] deployed CMV interfaces to tablets, but focused on displaying multiple visualizations on a single device.

The **combination or arrangement of mobile displays** can be used to, e.g., counteract the issue of limited display space when working with mobile devices. Several investigations have enhanced usability through cross-device interactions (e.g., Lucero et al. [35] used a separate device as controller) or through display area extension using side-by-side arrangements (e.g., forming a matrix or line [32,35,42]). These investigations differ in the way how they connect displays, for instance using touch gestures [35,42] or camera-based detection [32].

The field of **spatially-aware mobile displays** aims at using spatial relations and movements of those displays as input. On the one hand, the spatial position can be used to inform the system of the interaction context: Marquardt et al. [38] improved, e.g., the user's awareness of surrounding devices, and Ledo at al. [30] activated or deactivated appropriate functionalities depending on the spatial context. On the other hand, Spindler et al. [55] showed that the position and movements in space can also be used for zoom and pan interactions.

The **combination or arrangement of spatially-aware mobile displays** enables further interaction styles, e.g., by finer distinguishing the relative positioning. Marquardt et al. [39] focused on the collaborative use of such devices. By applying the principles of *f-formations* to spatial patterns of people's mobiles, the authors aimed to simplify data exchange. Lissermann et al. [33] investigated the use of relative spatial display movements for, e.g., controlling parameters. However, they concentrated on the mapping of a desktop-like interaction style to this type of displays. Piazza et al. [43] explored the specific combination of a smartphone and a tablet. The smartphone supports input and output modalities, e.g., for UI offloading or as a spatially-aware controller. Rädle et al. [44] mainly focused on the deployment of a low-budget tracking system for mobile devices, but also presented concepts for peephole interaction, data transfer, and UI offloading using multiple mobile devices. Later, Rädle et al. [45] showed that study participants tend to prefer spatially-aware over spatially-agnostic interaction techniques.

Spatially-aware mobile displays for InfoVis use spatial input as a modality of its own to, e.g., support specific InfoVis tasks. Spindler et al. [55] developed a basic spatial input design space using multiple tangible views above a tabletop (e.g., control of a detail view through 3D movements). They also found that for navigation tasks spatial input can even outperform established touch gestures [54]. Recently, Kister et al. [27] used spatially-aware tablets to explore graph visualizations on a large vertical display. Wozniak et al. [58] used 2D movements of a smartphone to control navigation within a visualization on a tablet which was set up on a table.

When looking at *multi-display environments* for InfoVis in general, several technical approaches or platforms exist. For instance, Hartmann et al. [17] introduced the concept of meta applications, which allow to run multiple instances of existing web applications. Monroe and Dugan's Disperse [41] allows to show multiple views on different screens by using simple client-side markups. Munin by Badam et al. [3] is a Java P2P framework for building ubiquitous analytics applications that use different input and output devices. Badam and Elmqvist's web-based PolyChrome framework [2] combines a P2P and client/server approach. While these approaches show how to display or synchronize views on multiple screens on a technical basis, our work instead focuses on visualization and interaction aspects. We specifically investigate the way of using a multi-display interface for data exploration.

Interestingly, the **combination of mobile displays for InfoVis** has rarely been investigated. To a certain extent, Hamilton and Wigdor's

[1]Prototype sources: https://github.com/imldresden/vistiles

Conductor [16] fits into this category, as it allows linking the content of different devices (e.g., highlighting addresses from a contact list in a map). However, neither the explicit arrangement of devices nor the interaction with visualizations is focused on. Similar, Fuchs et al. [10] and Langner et al. [28] investigated how to use the combination of multiple tangibles with small displays to query or control a data set. Kim and Elmqvist's embodied lenses [25] also focused on the physical affordance; they used thin transparent foils as lenses for visual queries. However, large parts of these works rely on using a separate context display (e.g., a tabletop). These concepts explicitly take advantage of such a display and are hard to use without it.

Noticeably, the existing work regarding InfoVis on mobiles mainly considered tablets as hardware and not smartphones, indicating that this research area is still underexplored. Furthermore, while CMV are well-established [47] and combinations of visualization views have been discussed (e.g., [23,36]), it still is an open question how to employ these concepts in a *ubiquitous visual computing space* [3]. Chung et al. [7] illustrated the potential of using *display ecologies* for visual analysis and also pointed towards the importance of physicality and space. In this work we take up on these insights and explore specific visualization concepts and their implications for CMV that are distributed across a set of co-located and spatially-aware mobile devices. The related research fields and their intersections discussed above provide essential concepts (e.g., the balance of spatially-aware and spatially-agnostic interactions [45]), which are relevant in the context of this work and informed the development of our conceptual framework. We will even show that our framework is capable of enabling and integrating existing concepts and techniques (e.g., [11, 23, 34]).

3 THE VISTILES FRAMEWORK

In this section, we describe the fundamental concepts and design of VISTILES. After briefly introducing the general idea (Sect. 3.1), we provide details about the two main components of the framework: *dynamic and individual layouts for CMV* (Sect. 3.2) and *smart adaptations and combinations of visualizations* (Sect. 3.3).

3.1 Conceptional Basis

The VISTILES framework builds on the idea of enabling CMV on multiple mobile devices, i.e., transferring a traditional visualization interface from a desktop-based environment to mobile devices. As a naive approach, one could just scale down a CMV application to the display size of a single mobile device. However, this only works to a certain degree and is limited by application complexity and number of visualization views. The VISTILES framework, instead, applies a 'divide and conquer' strategy: Individual visualization views of a CMV application are distributed across multiple mobile devices (Fig. 1). This provides an alternative visualization interface that permits a varying number of users to visually explore and analyze data with their mobile devices. Users benefit from the use of a *physical workspace*, which allows to grasp (pick up, tilt) and spatially organize visualization views.

To illustrate this idea, we briefly describe a potential setting: We envision that users sit at a table with multiple mobile devices. They want to analyze a multivariate data collection and set up their devices in a way that each device shows another visualization, i.e., different perspectives onto a data set. Similar to traditional systems, visualizations are linked with each other allowing, e.g., cross-device brushing: to share an interesting cluster of data items, a user can select these data items, whereby visualizations on all other coupled devices immediately highlight these items. Furthermore, users can move and reorganize visualizations on the table in useful and meaningful ways. On the one hand, they express certain intentions through the arrangement, e.g., put a device aside to 'come back to it later' or order visualizations to define a reading direction. On the other hand, the system also recognizes specific arrangements. For instance, by simply placing two visualizations side by side, the users can compare different visualizations or seamlessly synchronize filters, encodings, or configurations. These device combinations can of course also enable further functionalities, e.g., composing visualizations, extending screens, or offloading components of the user interface (such as menus or parameter windows). The arrangements, however, are flexible and can be altered anytime to support varying needs of diverse situations and scenarios.

This illustrates that VISTILES builds on both the distribution of visualization views across multiple devices as well as cross-device interaction techniques. As a prerequisite, it is assumed that all devices are connected and communicate via network as well as that side-by-side device arrangements can be detected. By knowing the visualization states on devices and which devices are arranged side by side, the application can actively support users by suggesting options to extend visualizations across screens or to filter data items from one device based on selections from another device.

3.2 VISTILES: Dynamic and Individual Layouts for CMV

As we suggest distributing CMV across several mobile devices, a *physical workspace* with graspable visualizations emerges. This enables users to construct and adapt their individual visualization interface in terms of positioning UI components. In order to describe how users can interact with this interface, we first specify aspects of the related design space. On this basis, we define different view types and their distribution, describe how they can be coordinated using workspaces, and finally illustrate the benefits of spatially organized visualizations.

3.2.1 Design Space: Input and Output

Based on prior work and our own explorations, we summarize essential input and output aspects when working with several mobile devices.

Surface Interactions Direct interactions on mobile devices represent the primary input channel, i.e., by means of direct touch, multi-touch, pen, or even tangible input. Elements and functions of the visualization can be manipulated directly (e.g., [4, 8, 48]). For the application domain of InfoVis, direct interaction already provides a tool to reduce the extensive use of complex menus and dialog windows [8].

Technical Properties Mobile devices have different characteristics that can influence the way how they are or are not used. In the context of InfoVis, display size, resolution, and pixel density play an essential role. For instance, readability is hard to guarantee on small or low-res devices. Also, the aspect ratio can for example affect the suitability regarding different visualization techniques. Other properties include weight, thickness, location of physical buttons, and bezel size.

Contextual Properties Most importantly, mobile devices that are placed on a table have three basic degrees of freedom: a position on the surface (x, y) and an orientation (α). The number of devices also plays a crucial role. Whereas two devices are the minimum for our concepts, a larger number allows more visualization views, provides larger screen estate, and supports collaboration. Finally, for multiple devices the distance, relative position, and orientation can be considered.

3.2.2 Tiles: View Types and View Distribution

As we load visualizations onto mobile devices, i.e., flat and rectangular artifacts that can be arranged in relation to each other, we call them **vis**ualization **tiles**. We designed our concepts in a way that for each tile users can assign a specific CMV component. By distributing these components across tiles, VISTILES reduces the interface complexity per tile and maximizes the size of each visualization. Based on common CMV, we propose to distinguish between two general types of tiles: (i) *data tiles* are used to display a visual representation of data, i.e., show data using a specific visualization technique; and (ii) *control tiles* display other elements of the UI such as menus or widgets, which are used for additional functionalities (e.g., changing visualization parameters or dynamically querying data items [52]). Despite this distinction, the variety of existing visualization techniques and their individual configuration possibilities justify that data tiles can also display on-screen menus or controls locally.

Furthermore, we suggest supporting the distribution process (i.e., selecting the type of a tile and specifying a visualization technique) by considering technical device properties. Large devices such as tablets are particularly qualified for visualizations (data tile), whereas small devices such as smartphones are well-suited for displaying menus or other UI elements (control tile). Besides the display size, the aspect

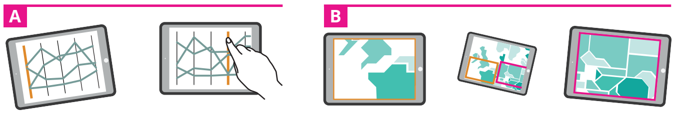

Fig. 2. Coordination of visualizations in workspaces: (a) highlighting shared axis by touch (extended linked brushing); and (b) indicating position and size of other viewports for overview & detail.

ratio makes devices suitable for certain visualization techniques. Wide or tall visualizations such as parallel coordinate plots or population pyramids benefit from narrow screens (wide-screen or 16:9). Radial visualizations typically use square areas; those can be better displayed on devices with a 4:3 ratio. As a result, a system which recommends the type of a tile or a specific visualization technique based on technical device properties can (i) reduce interaction costs and (ii) support users, particularly inexperienced users or people who are not aware of possible advantages or disadvantages of specific view distributions.

3.2.3 Workspaces: Coordination of Visualizations

Visualizations of CMV are typically linked (coordinated) and user interactions within one of the views also affect other views. In the VISTILES framework, a set of tiles that are coordinated is called *workspace*. In addition, VISTILES allows users to work with multiple visualization workspaces. The smallest possible workspace consists of a single tile; other tiles can be added subsequently (join). A tile can also be removed from a workspace (leave). As default behavior, we suggest that new tiles automatically join a main workspace. In general, workspaces are a way to easily control (activate or deactivate) coordinations between different tiles or visualization views. Also, they ensure flexibility for typical multi-user situations, where data analysts join or leave ongoing working sessions. While prior work reported on further issues or design tradeoffs of collaborative work (e.g., [14, 21, 37]), VISTILES focuses on enabling basic collaboration (cf. workspace coordination and collaboration styles [19]).

In the following, we illustrate how the concept of workspaces addresses three common InfoVis tasks or interactions: linked brushing, filtering data, and overview & detail. Besides these basics, other more complex linking mechanisms exist, for example: Isenberg and Carpendale [19] used linking to facilitate hierarchical data comparison tasks; and Tobiasz et al.'s Lark [56] even links views regarding the analytical abstraction, visual representation, and presentation. If needed, such mechanisms can be smoothly integrated in VISTILES.

Linked Brushing Consistent with CMV, selections and highlights are synchronized across data tiles. Additionally, we propose extending linked brushing to other visualization elements. For instance, when an axis of a parallel coordinates plot is touched, other tiles of the same workspace also highlight this axis (Fig. 2a). This is particularly useful as visualizations can be spatially separated. However, interactively highlighting other elements of both the visualization and the UI can support users in realizing similarities or differences between tiles.

Filtering Data In addition to common data filter possibilities (e.g., using UI elements such as sliders), we introduce a 'filter-by-viewport' mechanism. The idea is to "change the set of data items being presented" [59] on tiles of a workspace by zooming or panning, thus changing the viewport of one of these visualizations. For example, when zooming and panning a scatterplot, coordinated tiles immediately filter data items that are not visible (offscreen) in the scatterplot (Fig. 1e). As a consequence, data analysts can activate this additional filter mode for tiles that display a zoomable visualization individually by using a menu option.

Overview & Detail We also support the concept of overview & detail. When two or more tiles show a subset of the same data space, they automatically indicate the position and size of other viewports by displaying corresponding bounding boxes (Fig. 2b). This concept also allows a remote manipulation of such views by interacting with the bounding boxes. In order to avoid interaction conflicts (e.g., when two users zoom or pan a view simultaneously), tiles can be locked (disable remote manipulation) through a menu option. Moreover, the identification of specific overview & detail tiles is supported by temporarily displaying notifications (e.g., flashing screen) when bounding boxes are touched. This is especially important in cases where multiple tiles are part of such an overview & detail configuration.

3.2.4 User-defined Layout: Organizing Tiles

Since CMV components are distributed across multiple mobile devices, VISTILES enables analysts to take advantage of a flexible, dynamic, and user-defined arrangement. There is no fixed or immutable interface, instead it can be adapted according to the requirements of different situations. While the interface supports collaborative and multi-user usage in particular, even single users can benefit from this. With this flexibility, VISTILES addresses the concept of the "intelligent use of space" by Kirsh [26] and "space to think" by Andrews et al. [1]. Most interestingly, Kirsh notes that "whether we are aware of it or not, we are constantly organizing and re-organizing our workplace to enhance performance" [26]. As visualizations become graspable through the use of mobile devices, thus introducing an affordance, we aim to exploit this natural behavior for InfoVis.

In this sense, VISTILES allows users to physically organize and arrange tiles in useful or meaningful ways, which can provide both "external memory and a semantic layer" [1]. To illustrate some possibilities, tiles could be spatially arranged so that: (a) the location of a tile reminds people to come back to it later; (b) the sorting encodes a specific reading direction or story; (c) certain locations are associated with specific information to support rediscovery; (d) the location indicates groups or clusters of data items; (e) the sorting reflects the realized analysis process or history; or (f) the displayed content is clearly visible and readable by multiple users, for example in data presentations.

3.3 VISTILES: Adapting and Combining Visualizations

In the previous section, we described how to use physical device arrangements in a passive way (i.e., the system itself does not use this information). In this section, however, we propose to utilize the physical device proximity explicitly. We first summarize relevant aspects of the input design space. Then, we present concepts that directly use *spatial side-by-side arrangements* and *continuous device movements* to enable an interactive adaptation and combination of visualization views. Finally, we describe our approach of providing users full control regarding the selective activation of such adaptations and combinations.

3.3.1 Expanded Input Design Space

In order to inform the design of the concepts in this section, we expand the previously presented design space (Sect. 3.2.1) by two additional input aspects: side-by-side arrangements and spatial movement.

Side-by-side Arrangement Devices placed on a table can be arranged in various ways (e.g., [32, 35, 43, 44]). However, we found that in particular the direct side-by-side arrangement of two devices represents a very explicit user interaction. This arrangement depends on the contextual properties (Sect. 3.2.1) and is specified by two simple attributes: (i) the direction of alignment (i.e., horizontal or vertical); and (ii) the orientation of devices (i.e., landscape or portrait mode and any combination thereof). Even though we limit the use of spatial arrangements to side by side, combinations of more than two devices can result in potentially complex display arrangements and layouts.

Spatial Movements Besides device arrangements, spatial input with a device can also be used for input. Although it can be performed in 3D (cf. tangible views [55]), we again simplify input by limiting spatial movement to 2D movements in favor of a reduced complexity. Thus, devices can be moved or rotated absolutely (in relation to the physical world, e.g., the table) or relatively (with regard to another device). Basic types of 2D device movements (e.g., [55, 58]) include linear paths (along the edge or away/towards other devices), circular movements (around a device) [43], rotations, and combinations of those movements (e.g., vertical and horizontal). Moreover, the type of movement can also be gestural (e.g., shaking or flipping).

Fig. 3. Alignment and rearrangement: (a) aligning visualizations and rearrange axes; (b) merging data items of two bar charts and displaying computed data on the attached device; and (c) creating a scatterplot matrix by sequentially attaching devices.

3.3.2 Use of Side-by-Side Arrangements

In the following, we describe smart visualization adaptations that can be enabled by the side-by-side arrangement of tiles and actively support common InfoVis tasks or goals (e.g., visual comparison, details on demand, identify outliers, reveal correlations and distributions). These seven concepts range from slight changes to more complex combinations of visualizations.

Alignment As a simple but powerful concept, we recommend aligning visualization that are displayed on apposed tiles. By automatically translating, rotating, or scaling the views of tiles, they adapt to the properties (e.g., resolution, pixel density, position, orientation) of their counterpart. This ensures that visualizations and their elements are visually aligned and use the same physical scale, for instance, by adapting the length of axes (Fig. 3a). Although visualizations can become smaller and display space can remain unused (Fig. 3a), the alignment increases readability across tiles and counteracts misinterpretations caused by differences in size or shifted visualizations.

Rearrangement In addition to the alignment, we provide a concept for the rearrangement of visualization elements, such as data items or axes. For example, when two bar chart tiles are combined vertically, the sorting of data items (bars) could be equalized. Here, the comparison of displayed values is supported by reducing the visual gap (distance) between visualization elements. Moreover, we suggest using the rearrangement to provide additional, computed information (Fig. 3b). The idea behind this is to merge data items from two tiles (e.g., two bar charts) within one of these tiles. Then, the other tile is used to display calculated data such as differences (cf. explicit encoding [12]). Again, this eases data comparison but at the same time supports further exploration (cf. explore [59]).

In addition, VisTiles also uses a *combination of alignment and rearrangement*, by turning tiles into one continuous visualization where appropriate. For instance, a set of side-by-side scatterplot tiles is automatically turned into a scatterplot matrix (Fig. 3c). This involves both the alignment of the plots as well as a possible rearrangement of their axes. Alternatively, apposing a scatterplot tile and a tile showing a parallel coordinates plot (pc plot) results in the following three visualization changes (Fig. 3a): (i) aligning the scatterplot to the pc plot, (ii) rearranging the pc plot's axis that corresponds to the scatterplot's y-axis (here moved to the right), and (iii) highlighting both dimensions (i.e., axes) of the scatterplot in the pc plot. These changes and the resulting continuous visualization can support data comparison and can enable users to recognize shared dimensions more easily.

Adaptation of Encodings Besides adjusting the visual appearance of a tile by using a menu or other UI elements, we suggest sharing such encodings of visualizations across multiple tiles. For example, when a bar chart tile is placed side by side to a tile displaying a map visualization, the bars are re-colored according to the map's color scheme or vice versa (Fig. 4a). Additionally, scales of the axes can be synchronized, for example, in case of combining two scatterplot tiles. This is useful when, e.g., analysts have different personal preferences or a data analyst joins an ongoing working session with new devices.

Display Extension Another concept in the context of visualizations on mobile devices is to increase display size by side-by-side device combinations [32, 35, 42], i.e., to extend display space by 'stitching' multiple tiles (Fig. 4b). This expands a visualization across the screens of tiles. Similar to alignment and rearrangement, we suggest considering technical device properties such as pixel density. This counteracts misinterpretations caused by differently sized visualizations.

Dynamic UI Offloading In addition to using a device as a workspace-wide control tile, VisTiles allows dynamically offloading UI components of visualizations. A side-by-side arranged control tile automatically displays corresponding controls of the other visualization tile, such as data mapping, encodings, or color scheme (Fig. 1b). Besides tile properties, additional or detailed information about selected data items can be displayed (Fig. 4c, details on demand). Alternatively, a control tile can also access and manipulate the properties of other tiles from the workspace remotely. However, as the workspace can consist of many tiles, users have to select the corresponding tile from a menu. In general, the concept of UI offloading maximizes the size of visualizations and reduces visual clutter caused by on-screen menus.

Extended View Synchronization Besides the classic linked brushing within a workspace, the side-by-side arrangement can also be used to initiate a 'stronger' coordination of tiles. In the sense of an additional linking level, the extended view synchronization coordinates user interactions beyond the selection of data items or brushing between multiple tiles simultaneously. Depending on the specifics of involved visualizations, different effects or results are useful. For example, a side-by-side arrangement can enable the 'filter-by-viewport' mechanism described above (Sect. 3.2.3), which transforms zoom and pan interactions into filter queries. Taking the example of network visualizations, Figure 4d shows a node-link diagram arranged side by side with a corresponding adjacency matrix, allowing an easier exploration and manipulation of relations (cf. Gladisch et al. [11] and Kister et al. [27]). In case of map visualizations, the extended view synchronization can be used to overview and navigate (zoom and pan) multiple views at once. Overall, users can benefit from this concept as it reduces the effort for actions that should be applied to multiple tiles. It also exploits the concept of apposing devices that should share their states.

Supporting Advanced Exploration Techniques While the VisTiles framework mainly focuses on CMV, other state-of-the-art visualization concepts can be integrated and supported as well. For example,

Fig. 4. Further visualization adaptations: (a) adapting the color scheme; (b) extending displays and thus visualizations over two devices; (c) offloading details of selected data items; and (d) editing relation of a node-link diagram through the corresponding adjacency matrix.

the framework can implement *composite visualization views* (CVVs) proposed by Javed and Elmqvist [23]. When two data tiles are arranged side by side, corresponding superimposed, overloaded, or nested views can be generated. While juxtaposed views are directly integrated into our framework, the usefulness of integrated views particularly depends on the bezel sizes of the used devices, as they rely on “explicit visual linking between multiple component visualizations” [23]. Furthermore, the concept of *heterogeneous embedded data attributes* (HEDA) by Loorak et al. [34] can also be applied directly. For that, a data tile representing the interactive tabular visualization component HEDA can be ‘attached’ (side-by-side arrangement) to other tiles in order to display “multi-dimensional data details” [34].

These examples illustrate that the VISTILES framework can also provide a visualization interface for techniques enabling advanced data exploration by using side-by-side arranged visualizations.

3.3.3 Use of Continuous Spatial Movement

The adaptations described above build on explicit and rather static side-by-side arrangements of tiles. By arranging tiles, a certain functionality can be activated as a discrete event. As an extension to this, tile combinations can additionally make use of continuous device movements. When two tiles are already arranged side by side, one tile can be moved along the other tile. Similar to Wozniak et al. [58], we suggest mapping spatial movements to a selected visualization parameter or data dimension and thereby allow users a continuous manipulation.

A basic example is the manipulation of a slider widget that allows dynamically filtering displayed data by year. For that, a smartphone and a tablet are arranged side by side. As a data tile, the tablet shows data items regarding two data dimensions in a scatterplot. The data is available as a time series, thus users can also select a specific year for the displayed scatterplot. Here, the smartphone (as a control tile) represents a physical slider to manipulate the displayed year. Now, instead of touching a slider widget, the smartphone can be moved along the scatterplot tile (in this case vertically) to select a year and thereby adjust the scatterplot (Fig. 5a).

Furthermore, the spatial movement is also useful in an overview & detail configuration. Fig. 5b illustrates an example where one tile shows a more detailed but smaller area of another tile displaying a timeline visualization or line chart. Again, instead of touching the devices and occluding information, users can pan the detail view by physically moving the tile along the timeline (here horizontally).

The concept of using continuous spatial movement is designed as an alternative to surface interactions such as dragging a slider. Besides enabling user interactions that do not affect the visibility of information negatively, this is also motivated by people’s dexterity—the exceptional ability to grasp and manipulate real objects with the hands. This skill allows us to interact with objects without looking at them. In addition, associating abstract variables or data with real physical locations of objects can help to find certain information again (cf. recall [26]).

3.3.4 Managing Adaptations and Combinations

The VISTILES framework features a set of different adaptations and combinations of visualizations. As there are several possible system reactions based on a specific arrangement, the question arises what adaptation or function is activated in which situation. In general, two strategies can be applied: (i) The system suggests or recommends a set of useful options, which can be activated by the analysts; or (ii) the system activates the most ‘appropriate’ option and the user can correct this afterwards. For the VISTILES framework, we propose to use the first strategy—‘the application suggests, users confirm’. When moving tiles towards each other, possible suggestions include: extending a visualization, aligning a view, filtering data items, or enabling the synchronization of zoom and pan between two tiles. By offering such options, we aim to avoid that users feel irritated or even lose control by sudden changes of the visualization. In this context, it can also be helpful to animate or cross-fade such visual changes.

In addition to the suggestion and activation of options, the direction of an adaptation is important. The example of combining two bar chart tiles (Fig. 3b) illustrates that it is not clear which tile shows merged data items and which displays additional information. Therefore, we propose to present options on each tile that is involved (Fig. 6d). Then, users apply the action to a specific tile by activating the corresponding option on that device. However, for some cases it is sufficient to consider the type of a tile: Since the control tile is attached to a data tile, it is the control tile which shows details of selected data items or offloaded user interface elements of the corresponding data tile, not vice versa. Finally, we also suggest employing these options to deactivate previously applied adaptations, instead of moving a tile. Again, this gives users control and allows for example to pick up a device from a side-by-side arrangement for flexibility or comfort.

As a result of using information about involved tiles (e.g., type, displayed information, horizontal or vertical), the presented approach of suggesting possible adaptations and combinations of visualizations provides a careful balance between user control and system control.

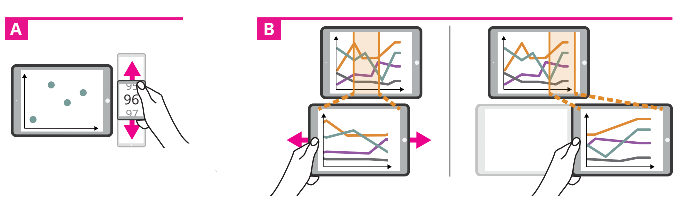

Fig. 5. Using continuous spatial movements for manipulating (a) visualization parameters and (b) overview & detail views.

4 REALIZING VISTILES: PROCESS AND PROTOTYPE

In the previous section, we presented the fundamentals of the VISTILES framework on the conceptional level: views distributed across devices, device layout and arrangement, as well as visualization adaptations and combinations. Now, we explain the development process to provide a better understanding of how these concepts evolved and to report on our functional software prototype, which realizes our concepts. The purpose of this prototype is to allow the demonstration, discussion, and evaluation of the concepts. The structure of this section corresponds to the three main phases of our iterative development process: first we focused on a *proof of concept* and the initial design, we then conducted a *preliminary user study*, and finally we *revised the concepts* and enhanced the prototype.

4.1 Phase I: Proof of Concept

In the beginning, we focused on the general *design and prototyping*. We first elaborated initial ideas and principles as well as showcased those using paper prototypes and a conceptual software prototype. Then, we discussed with colleagues about different variants of both interaction and visualization concepts to identify promising approaches as well as challenges [29]. To rapidly test our ideas, the conceptual software prototype used simple UI mockups (static images) and scripted transitions. These transitions simulated system reactions on mobile devices by blending in components of the UI. Although simple, the prototype already processed proximity events based on the physical distance between devices.

4.2 Phase II: Preliminary User Study

In the second development phase, we aimed to gather *early user feedback* to validate the fundamental functionality and feasibility of our concepts. Therefore, we developed a second, more detailed working prototype that builds on modern web technologies. By using this web-based prototype, we then collected qualitative user feedback to learn about, e.g., which visualization adaptations are considered useful, and for which purpose people would like to use device arrangements. In general, we wanted to explore explicit opportunities for improvements of both the VISTILES concepts and the prototype.

4.2.1 Web-based Prototype: Setup and Functionality

The technical setup of our prototype consists of a set of mobile devices (Android-driven, 1× 5” Google Nexus 5, 2× 7” Google Nexus 7, 2× 8.4” Dell Venue 8) and an external camera-based tracking system.

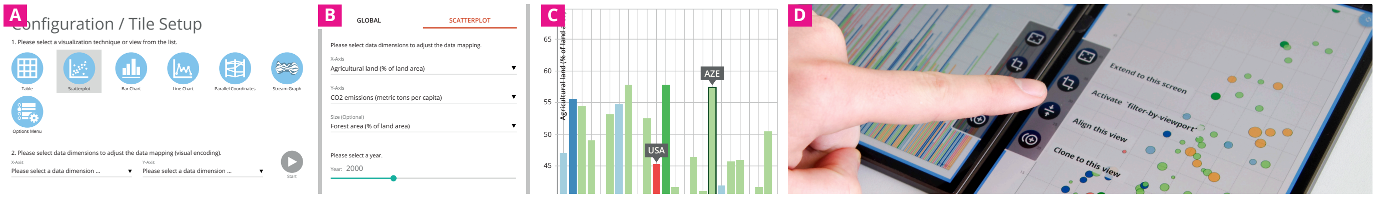

Fig. 6. Realization of our web-based prototype: interface design of (a) the main menu, (b) a control tile, and (c) a data tile, here a bar chart with multiple selections. (d) Managing visualization adaptations through user options displayed at the borders of side-by-side tiles.

The tracking system[2] is installed above a table. In order to track the positions of devices, we equipped them with markers using dark IR-reflective foil (unobtrusive for users). The desk was waist-high, thus easy to use while standing, and fully covered by the tracking system.

Similar to other cross-device research prototypes (e.g., [2,44,51]), we decided to build on modern web technologies in order to support the majority of mobile devices. To provide, manage, and drive communication between involved devices, we realized a client-server-architecture. On the server side, a Node.js[3] server application handles client communication via WebSockets. It consumes the tracking data to detect proximity-based device combinations and controls corresponding system reactions. On the client-side, the browser connects and loads the UI. We use D3.js[4] to create visualizations and the Materialize CSS framework[5] for a consistent user interface.

As example data, a subset of the *WDI*[6] time series data collection was used, containing 52 dimensions for 215 countries (data items) over 24 year (1991 to 2015). In this second phase, the prototype contained three simple visualizations: bar chart, line chart, and scatterplot. The spatial arrangement was used to automatically create workspaces; the side-by-side adaptations were directly applied without suggesting options to users. Depending on the combined visualizations, predefined adaptations were activated: alignment, rearrangement, display extension, 'filter-by-viewport', or overview & detail.

4.2.2 Study Design

The preliminary user study aimed at testing our assumptions regarding the spatial device arrangements and combinations for data exploration and that our concepts are feasible to support this exploration. The study was conducted with the web-based prototype described above.

Participants Seven unpaid students from the local university (age M= 24.43, SD= 2.37 years; 3 female; 5 post-graduated) volunteered for the user study. One student was from the department of psychology, the remaining from the computer science department. All participants had basic knowledge of information visualization (e.g., through courses) and two already worked with visualization software such as Tableau[7]. Also, all of them owned a smartphone and three used tablets regularly.

Procedure Each feedback session lasted 45 minutes and was conducted as a semi-structured interview. First, we introduced the basic idea of VisTiles, then we sequentially demonstrated and discussed examples of our concepts: (i) dynamic UI offloading with a control tile, (ii) display extensions with a bar chart, (iii) filtering data based on the viewport of scatterplot ('filter-by-viewport'), (iv) overview & detail with line charts, and (v) alignment and rearrangement concepts. Participants were asked to try out the concepts by themselves. For each example, participants interacted with the prototype and provided feedback (think aloud and verbally answer questions). Sessions ended with a discussion on overall, more general aspects regarding our concepts. Two researchers took notes (one of them guided the interview) during the complete session, which was also videotaped.

[2]OptiTrack by NaturalPoint, http://optitrack.com/
[3]http://nodejs.org/ [4]http://d3js.org [5]http://materializecss.com
[6]World Development Indicators, http://databank.worldbank.org/
[7]https://www.tableau.com/

4.2.3 User Feedback

Use of Spatial Arrangements In general, all participants appreciated that the interface incorporated the spatial arrangement—regarding both the natural interaction style with mobile devices as well as the system-supported cross-device functionalities. Especially the side-by-side arrangements were rated as an intuitive (P1) and logical (P2, P3, P6) way of interaction similar to known workflows (P3: "moving devices around is analogous to working with paper"). However, P5 and P7 expressed concerns regarding the enormous use of spatial movements in order to activate functions such as joining/leaving a workspace based on the distance between devices. Especially when using many devices within a limited space, the distance-based thresholds can activate functions accidentally (P3, P4), a functionality that has been discarded in the final concepts. P4 also noted that "it is hard to remember all possible arrangements that are assigned to specific functions" (hidden functionalities). Summing these comments up, participants felt that in contrast to side-by-side arrangements, the device distance is not applicable to activate systems reactions automatically.

Interaction Concepts The offloading of UI components such as tile properties to a separate device was positively assessed by all participants. Three also commented favorably on the explicit side-by-side arrangement for this concept (P1, P3, P6). In addition, P4, P6, and P7 suggested to permanently use a dedicated settings device, making it possible to adjust parameters of multiple visualizations from a distance by, e.g., selecting a target device from a list. In this case, the side-by-side arrangement could be optional and act as "a shortcut" (P6). For the display extension example, we asked participants if they found it intuitive that the horizontal combination automatically resulted in such an extension. Although P7 commented "it is super useful and intuitive", most participants were more critical: They rated it less intuitive (P3, P5), somehow arbitrary (P1, P5), and challenging when different display properties are involved (P2, P5). This shows that there is a need to explicitly manage, which adaptations should be applied. The filter-by-viewport example, i.e., filter other visualizations by zooming and panning a scatterplot (Fig. 1d), was stated as the "coolest" (P3, P6) example (supposably as it illustrates the powerful possibilities of our concepts). Again, P4, P6, and P7 suggested allowing this filter functionality at a distance. Finally, in the overview & detail example all participants used the possibility to interact on both the overview and the detail view to manipulate the detail view's viewport. Comments focused on personal preferences that, e.g., working on the overview is preferred (P6) or only using it when tiles are close (P2).

Workflow As already mentioned, the implemented examples also initiated discussions about the workflow for managing visualization views, i.e., when to automatically apply changes and when to restrict it to manual, more explicit user actions. Some system reactions were not considered as intuitive (e.g., display extension) or desirable (e.g., creating workspaces based on device distance) to all participants. Therefore, four participants (P1, P4, P5, P6) proposed to only offer the user possible adaptations, which then can be selected manually. Further, participants wanted to be in control of how long an adaptation is active. For instance, P6 commented that filtering the bar chart through a scatterplot should still be possible when the scatterplot device is picked up, i.e., the side-by-side arrangement is repealed. Finally, participants (P1, P3, P4) also distinguished device roles based on the device size, e.g., that the smartphone is more appropriate for settings or filter views.

4.3 Phase III: Framework Revision and Final Prototype

The third development phase comprised the revision of the framework considering the experiences and results from previous phases. We analyzed feedback from the preliminary user study and discussed with experts about improvements of our concepts and how to further ease data exploration. The concepts presented in this paper already reflect the result of this third phase. Correspondingly, we also improved and extended the web-based prototype from the second phase.

After reporting on lessons learned and revised concepts, we inform about the components of the final prototype, in particular the realized interface design. Finally, we illustrate a data exploration process with VISTILES by describing an interaction walkthrough.

4.3.1 Lessons Learned and Revision of Concepts

First of all, user feedback from the second phase confirmed that *device movements need to be used carefully for activating changes*. As such movements might happen during other actions with mobile devices (e.g., taking a closer look at a display), unintentional system reactions could be the consequence—similar to problems with gestural interaction. Therefore, we limited the use of spatial device input to side-by-side arrangements and discarded the dynamic creation of workspaces based on the device distance. As a result, users have full control of the way how the available physical space is used. In contrast, the direct side-by-side arrangement of devices is, indeed, perceived as a specific and explicit interaction by users. Still, alternatives should be provided to *activate certain functions even from a distance* (e.g., through menus) without the need of placing a device side-by-side.

Based on device arrangements, we initially envisioned several means of automatic adaptations. As stated by participants, automated changes are often challenging regarding accidental activations as well as concurrent possibilities. In most cases, multiple options of how visualizations can be adapted to each other exist. Therefore, we introduced the possibility to manage these adaptations (i.e., activate them manually). This avoids possibly irritating changes as users have full control regarding the applied adaptations.

The handling of visualization views, particularly the order in which they are created and analyzed, might affect the number, type, and quality of insights. Therefore, the VISTILES concepts focus on *enabling dynamic display configurations* with a *smart view distribution across mobiles* that support flexible workflows. On the one hand, the combination of tiles allows to dynamically increase display space (e.g., display extension, visualization adaptation) and thus addresses the issue of limited display size of single mobile devices for data exploration. On the other hand, the user feedback also confirmed that the number of devices and their properties have to be considered for a useful view distribution. Comments showed that offloading UI widgets is a very simple, but powerful technique. We found that both the distribution of views as well as the combination of devices support the user to get different perspectives onto the data and to easily adapt the interface to the current needs.

4.3.2 Components of the Final Prototype

Main Components The *main menu* is the entrance point into the application (Fig. 6a). A list of options allows users to choose between several visualization techniques as well as UI widgets, thus to assign the role of a data tile or control tile to a device. The control tile features a tab-based menu offering control elements such as sliders or drop downs for connected data tiles (Fig. 6b). A floating *menu button* displayed in the top right corner of each tile (see Fig. 1 and Fig. 6) can be used to return to the main menu at any time. Furthermore, the system provides *notifications* for important events or changes. For instance, in case of an applied adaptation between two tiles, each screen confirms this to the user by flashing and displaying a link icon. When multiple workspaces are used, the association of tiles is indicated by *colored borders* (one color per workspace). The color of the menu buttons is adapted correspondingly to strengthen this effect.

Visualizations In the prototype, we implemented a number of visualizations including bar charts, scatterplots, line charts, parallel coordinate plots, stream graphs, and tables (spreadsheets). In all of them, a 10-class qualitative color scheme from ColorBrewer[8] is used to indicate data groups (Fig. 6c,d). Data items can be directly selected by touch. As a result, a small tool tip appears showing details such as the name (Fig. 6c). Currently, the scatterplot and line chart are zoomable.

Adaptations To avoid sudden and irritating user interface changes caused by side-by-side arrangements, the VISTILES prototype provides specific visualization adaptations as options (Sect. 3.3.4). These are listed in an icon bar (drawer), which appears at the border of arranged tiles (Fig. 6d). As soon as one device is moved away, the option menus slides back in and offers to disable the former selected adaptations or configurations. However, the user can also decide to keep synchronizations active and to manually deactivate them later. In the latest prototype we implemented the following side-by-side combinations: alignment and rearrangement, display extension, dynamic UI offloading, view synchronization for overview & detail, and 'filter-by-viewport'.

4.3.3 Interaction Walkthrough for Data Exploration

Goal of the following interaction walkthrough is to illustrate the application of VISTILES as well as to provide a better understanding of how data analysts can use such an interface for exploring multivariate data.

A group of three people wants to investigate a multivariate data collection collaboratively. They work together in a meeting room with a large table and numerous devices with different screen sizes. In the beginning, each user performs a quick *initialization* with some devices. This initialization of the mobile devices follows two steps: (i) By opening the systems' default browser and connecting to the server, all devices are paired with the systems, i.e., they are mapped to a corresponding ID of the motion tracking system. Since the server stores the identification mapping permanently, the pairing step can be skipped in future working sessions. (ii) Devices switch to full screen mode and, finally, load the start menu of the visualization application (Fig. 6a).

Since two of the three users are interested in particular data dimensions, they directly cooperate, use the shared main workspace, and focus on data visualized in a bar chart as well as in a line chart. They create a bar chart regarding urban population growth of the year 2007. The chart reveals that some data items have significantly higher values; identifying the State of Qatar and the United Arab Emirates as outliers. To further investigate this, they take a second tile and load a line chart showing for both that this is not a linear trend over the last 15 years, but a peak of a fluctuation.

The third user, however, first wants to investigate the data independently. Therefore, she takes one of the tiles and activates a 'create new workspace' option using the tile's floating menu button. Tiles can be assigned to workspaces by using either a tile's main menu or the side-by-side arrangement. By arranging another tile directly at the side of the first tile, a menu for managing adaptations appears at the corresponding border (Fig. 6d). One option ('join workspace') from this menu allows to add the tile to the new workspace in order to explore data using different views. Contrary to the others, this data analyst mainly uses scatterplots to focus on distributions and correlations in the data. At some point, she discovers a correlation in the data between the income per person and the child mortality rate. To share her insights, she adds the yet separated tile into the other workspace and moves this scatterplot towards the line chart. They select the previously identified outliers and realize that, contrary to the overall correlation, the State of Qatar and the United Arab Emirates have almost equal child mortality rates but quite a different income per person. Moreover, due to the side-by-side arrangement, the 'filter-by-viewport' option can be activated. Through filtering the line chart, they discover that especially for African states the child mortality rate decreases, while the urban population growth increases. By zooming and panning the scatterplot, they can further explore if this correlation also exists for different clusters.

5 LIMITATIONS AND DESIGN ALTERNATIVES

As indicated by related work (e.g., [1, 26]) and observed in our study, incorporating physical arrangements into the data exploration is indeed

[8] http://colorbrewer2.org

a useful approach. The VisTiles framework builds on this: By introducing a new way to handle multiple visualization views displayed on mobile devices, a co-located collaborative analysis with flexible workflows is enabled. However, there exist some limitations as well as design alternatives (e.g., interaction techniques, technologies), which we discuss in detail in the following.

Limitations One technical limitation is the requirement of a tracking system for localizing the mobiles devices, preventing an ad-hoc usage of our interface. For our prototype, we initially chose such an external tracking system as it offered accurate and stable position data and was easy to deploy into our implementation. However, as the revised VisTiles concepts presented in this work mainly build on side-by-side arrangements, it can also be implemented with a minimal extension of current mobile devices: Besides low-cost tracking systems (e.g., [24,44]), the most interesting solutions use lightweight localization approaches and internal device sensors only (e.g., [13,18,40]).

The preliminary user study has already provided insights into the use of spatial device arrangements, interaction concepts, and workflows. However, an in-depth study of the VisTiles concepts is needed to improve our understanding regarding the use of spatially-arranged mobile devices during a data exploration. As a result of the preliminary user study, we identified further questions (e.g., concerning concepts around "space to think") that should be investigated in future in-depth studies. This can involve the use of space in both single-user and multi-user scenarios. Also, deploying the interface to different data collections could provide further insights: People working with these collections might have other workflows and requirements that need to be considered. Finally, quantitative evaluations regarding the performance of a VisTiles system could help to better compare our approach to existing systems. Setting explicit data exploration tasks and comparing VisTiles with an existing application such as Tableau would allow to assess the utility of the presented concepts.

Divide and Conquer? Single-Display or Multi-Display When working with several visualization views on mobile devices, a key aspect is the way of how the views are displayed and distributed. In this context, the VisTiles concepts build on the strategy 'one visualization per device'. While this directly addresses issues of limited display space and visual clutter, it also requires the use of multiple mobile devices and thus introduces some challenges. When users bring their own devices, the ownership and role of individual devices have to be considered in social contexts. Perhaps, people prefer to only use their own devices or have concerns when others are using their private devices. Further, the complexity of possible combinations and spatial arrangements increases with the actual number of devices that are involved, but at the same time provides a certain flexibility.

As an alternative, a naive but obvious approach is to scale down a CMV application and display it on a (larger) mobile device that is capable of displaying the entire UI of an application. While this appears simple, Sadana and Stasko [49] showed that it also introduces some challenges. For instance, depending on the number of visualizations and UI elements, the presentation easily gets too small and thus cluttered. As a result, this would affect the readability and ability to touch visual elements for interaction. To counteract, the application can allow users to prioritize UI elements by minimizing, temporarily maximizing, or hiding these elements [49]. This, however, also increases the interface complexity regarding user interaction. Furthermore, when multiple users are involved, it clearly makes no sense that all of them have to share a single device. Although every user could use a single device that shows the entire application, some mechanisms and techniques need to be developed in order to support collaboration. VisTiles, however, distributes CMV components across multiple devices and enables that different users can work and interact with different parts of the application, thus supporting multi-user scenarios inherently.

Moreover, providing multiple devices per person enables the usage of a *physical workspace*. However, there is a need to further study the simultaneous use of mobiles devices, both for single-user and multi-user scenarios. Existing research reported differently on this: Whereas Hamilton and Wigdor [16] used up to ten devices and their spatial arrangement, Haber et al. [15] found that people tend to limit the usage to one device. In our feedback sessions, participants answered that four to seven devices might be manageable for a single user. Further research can help to clarify this specific aspect. In addition, user studies could formally compare benefits of flexible view arrangements to simpler means of interaction and view distribution.

'Space to Think': Physical Tiles or Virtual Tiles Another key aspect of VisTiles' *physical workspace* is the possibility to spatially arrange mobile devices and thus visualizations. This is different from desktop-based CMV, as (i) the arrangement is not limited to grid-based layouts and views can be moved freely, and (ii) views also become physical and tangible. The first characteristic can also be applied to other technical settings such as a large interactive display. Thanks to its size, such a display can show a large canvas with multiple virtual tiles (visualizations), which also supports multiple users and allows to freely move and arrange the tiles. A specific interaction style is, however, implied by the display orientation: horizontal displays (tabletops) suit face-to-face teamwork, while vertical displays support working from various distances (i.e., close for details, from a distance for overview).

The main differences between such a system with a large display and VisTiles is the usage of different tile types: physical or virtual. Obviously, virtual tiles on a large display can easily be created and manipulated (e.g., size, aspect ratio), thus they provide more flexibility and address applications scenarios where large data sets and large visualizations are used. However, mobile devices as physical tiles make use of people's dexterity as visualizations become tangible and provide physical affordance, which results in an easier handling (e.g., grabbing tiles without looking at them). Further, it is possible to spontaneously connect mobile devices at almost any place whereas large displays are rather stationary. In the context of evolving work environments, which provide a lot of space for flexible use, the need for ad-hoc data explorations is getting more prominent—a need that can be fulfilled with our approach. In conclusion, these two aspects, having physical visualization tiles as well as allowing ad-hoc device communities [44], are in our opinion two major advantages compared to static setups.

6 Conclusion

We presented VisTiles, a conceptual framework that builds on coordinated & multiple views that are distributed across multiple mobile devices. As these devices (called **vis**ualization **tiles**) can be easily grasped and moved around, this approach allows analysts to exploit a *physical workspace* in which they can freely arrange visualizations in useful and meaningful ways. We specifically support the exploration of multivariate data with concepts regarding: (i) The coordination and spatial arrangement of tiles in workspaces (e.g., linked brushing, filtering) to provide dynamic and individual layouts for CMV; and (ii) the adaptation, synchronization, and combination of visualizations in manifold ways based on explicit side-by-side arrangements of devices and spatial device movements along another device. This enables users to gather further insights regarding the data: We address common InfoVis tasks and goals with our concepts, such as the alignment and rearrangement of visualizations, extension of displays, or UI offloading. As illustrated by our prototype realization and feedback from a preliminary user study, VisTiles offers a promising approach for mobile InfoVis and provides a solution for a novel class of CMV systems.

Acknowledgments

The authors wish to thank: Ulrike Kister for valuable discussions and advice on the VisTiles concepts; Wolfgang Büschel, Annett Mitschick, Ulrich von Zadow, and other members of the Interactive Media Lab Dresden for their permanent support; the students Dennis Esders, Nils Henning, Daniel Fuchs, and Dennis Metzger for the help in implementing the final prototype; and finally, the anonymous reviewers for their thorough comments and suggestions to improve the paper. This work was supported in part by DFG grant GEMS 2.0 (DA 1319/3-3).

REFERENCES

[1] C. Andrews, A. Endert, and C. North. Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 55–64. ACM, New York, NY, USA, 2010. doi: 10.1145/1753326.1753336

[2] S. K. Badam and N. Elmqvist. Polychrome: A cross-device framework for collaborative web visualization. In *Proc. ITS '14*, pp. 109–118. ACM, 2014. doi: 10.1145/2669485.2669518

[3] S. K. Badam, E. Fisher, and N. Elmqvist. Munin: A peer-to-peer middleware for ubiquitous analytics and visualization spaces. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):215–228, Feb 2015. doi: 10.1109/TVCG.2014.2337337

[4] D. Baur, B. Lee, and S. Carpendale. Touchwave: Kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proc. ITS '12*, pp. 255–264. ACM, 2012. doi: 10.1145/2396636.2396675

[5] T. Büring and H. Reiterer. Zuiscat: Querying and visualizing information spaces on personal digital assistants. In *Proc. MobileHCI '05*, pp. 129–136. ACM, 2005. doi: 10.1145/1085777.1085799

[6] L. Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, March 2006. doi: 10.1109/MC.2006.109

[7] H. Chung, C. North, S. Joshi, and J. Chen. Four considerations for supporting visual analysis in display ecologies. In *IEEE VAST '15*, pp. 33–40. IEEE, 2015. doi: 10.1109/VAST.2015.7347628

[8] S. M. Drucker, D. Fisher, R. Sadana, J. Herron, and m. schraefel. Touchviz: A case study comparing two interfaces for data analytics on tablets. In *Proc. CHI '13*, pp. 2301–2310. ACM, 2013. doi: 10.1145/2470654.2481318

[9] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T.-J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011. doi: 10.1177/1473871611413180

[10] J. Fuchs, R. Rädle, D. Sacha, F. Fischer, and A. Stoffel. Collaborative data analysis with smart tangible devices, 2013. doi: 10.1117/12.2040011

[11] S. Gladisch, H. Schumann, M. Loboschik, and C. Tominski. Toward using matrix visualizations for graph editing. In *InfoVis '15*. IEEE, 2015.

[12] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 10 2011. doi: 10.1177/1473871611416549

[13] J. Grønbæk and K. O'Hara. Built-in device orientation sensors for ad-hoc pairing and spatial awareness. Workshop Paper at ISS 2016 Cross-Surface Workshop, 2016.

[14] C. Gutwin and S. Greenberg. Design for individuals, design for groups: Tradeoffs between power and workspace awareness. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pp. 207–216. ACM, 1998. doi: 10.1145/289444.289495

[15] J. Haber, M. A. Nacenta, and S. Carpendale. Paper vs. tablets: The effect of document media in co-located collaborative work. In *Proc. AVI '14*, pp. 89–96. ACM, 2014. doi: 10.1145/2598153.2598170

[16] P. Hamilton and D. J. Wigdor. Conductor: Enabling and understanding cross-device interaction. In *Proc. CHI '14*, pp. 2773–2782. ACM, 2014. doi: 10.1145/2556288.2557170

[17] B. Hartmann, M. Beaudouin-Lafon, and W. E. Mackay. Hydrascope: Creating multi-surface meta-applications through view synchronization and input multiplexing. In *Proceedings of the 2Nd ACM International Symposium on Pervasive Displays*, PerDis '13, pp. 43–48. ACM, 2013. doi: 10.1145/2491568.2491578

[18] K. Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pp. 149–158. ACM, New York, NY, USA, 2003. doi: 10.1145/964696.964713

[19] P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1232–1239, Nov 2007. doi: 10.1109/TVCG.2007.70568

[20] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K.-L. Ma, and H. Hagen. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization*, 10(4):310–326, 2011. doi: 10.1177/1473871611412817

[21] P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. *Computer Graphics Forum*, 28(3):1031–1038, 2009. doi: 10.1111/j.1467-8659.2009.01444.x

[22] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proc. CHI '08*, pp. 1217–1226. ACM, 2008. doi: 10.1145/1357054.1357245

[23] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *IEEE PacificVis Symp. '12*, pp. 1–8. IEEE, 2012. doi: 10.1109/PacificVis.2012.6183556

[24] H. Jin, C. Holz, and K. Hornbæk. Tracko: Ad-hoc mobile 3d tracking using bluetooth low energy and inaudible signals for cross-device interaction. In *Proc. UIST '15*, pp. 147–156. ACM, 2015. doi: 10.1145/2807442.2807475

[25] K. Kim and N. Elmqvist. Embodied lenses for collaborative visual queries on tabletop displays. *Information Visualization*, 11(4):319–338, Oct 2012. doi: 10.1177/1473871612441874

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

[27] U. Kister, K. Klamka, C. Tominski, and R. Dachselt. GraSp: Combining Spatially-aware Mobile Devices and a Display Wall for Graph Visualization and Interaction. *Computer Graphics Forum*, 2017. doi: 10.1111/cgf.13206

[28] R. Langner, A. Augsburg, and R. Dachselt. Cubequery: Tangible interface for creating and manipulating database queries. In *Proc. ITS '14*, pp. 423–426. ACM, 2014. doi: 10.1145/2669485.2669526

[29] R. Langner, T. Horak, and R. Dachselt. Towards combining mobile devices for visual data exploration. In *Poster Program of the 2016 IEEE Conference on Information Visualization (InfoVis)*, 10 2016.

[30] D. Ledo, S. Greenberg, N. Marquardt, and S. Boring. Proxemic-aware controls: Designing remote controls for ubiquitous computing ecologies. In *Proc. MobileHCI '15*, pp. 187–198. ACM, 2015. doi: 10.1145/2785830.2785871

[31] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *IEEE Trans. Vis. Comput. Graphics*, 18(12):2689–2698, 2012. doi: 10.1109/TVCG.2012.204

[32] M. Li and L. Kobbelt. Dynamic tiling display: Building an interactive display surface using multiple mobile devices. In *Proc. MUM '12*, pp. 24:1–24:4. ACM, 2012. doi: 10.1145/2406367.2406397

[33] R. Lissermann, S. Olberding, B. Petry, M. Mühlhäuser, and J. Steimle. Papervideo: Interacting with videos on multiple paper-like displays. In *Proc. MM '12*, pp. 129–138. ACM, 2012. doi: 10.1145/2393347.2393372

[34] M. H. Loorak, C. Perin, C. Collins, and S. Carpendale. Exploring the Possibilities of Embedding Heterogeneous Data Attributes in Familiar Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):581–590, Jan 2017. doi: 10.1109/TVCG.2016.2598586

[35] A. Lucero, J. Holopainen, and T. Jokela. Pass-them-around: Collaborative use of mobile phones for photo sharing. In *Proc. CHI '11*, pp. 1787–1796. ACM, 2011. doi: 10.1145/1978942.1979201

[36] S. MacNeil and N. Elmqvist. Visualization mosaics for multivariate visual exploration. *Comput. Graph. Forum*, 32(6):38–50, Sep 2013. doi: 10.1111/cgf.12013

[37] N. Mahyar and M. Tory. Supporting communication and coordination in collaborative sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1633–1642, Dec 2014. doi: 10.1109/TVCG.2014.2346573

[38] N. Marquardt, T. Ballendat, S. Boring, S. Greenberg, and K. Hinckley. Gradual engagement: facilitating information exchange between digital devices as a function of proximity. In *Proc. ITS '12*, pp. 31–40. ACM, 2012. doi: 10.1145/2396636.2396642

[39] N. Marquardt, K. Hinckley, and S. Greenberg. Cross-device interaction via micro-mobility and f-formations. In *Proc. UIST '12*, pp. 13–22. ACM, 2012. doi: 10.1145/2380116.2380121

[40] D. Merrill, J. Kalanithi, and P. Maes. Siftables: Towards sensor network user interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pp. 75–78. ACM, New York, NY, USA, 2007. doi: 10.1145/1226969.1226984

[41] M. Monroe and C. Dugan. *Disperse: Enabling Web-Based Visualization in Multi-screen and Multi-user Environments*, pp. 418–435. Springer International Publishing, 2015. doi: 10.1007/978-3-319-22698-9_29

[42] T. Ohta and J. Tanaka. Pinch: An interface that relates applications on multiple touch-screen by 'pinching' gesture. In *Advances in Computer Entertainment*, vol. 7624 of *Lecture Notes in Computer Science*, pp. 320–335. Springer, 2012. doi: 10.1007/978-3-642-34292-9_23

[43] T. Piazza, M. Fjeld, G. Ramos, A. Yantac, and S. Zhao. Holy smartphones and tablets, batman!: Mobile interaction's dynamic duo. In *Proc. APCHI '13*, pp. 63–72. ACM, 2013. doi: 10.1145/2525194.2525205

[44] R. Rädle, H.-C. Jetter, N. Marquardt, H. Reiterer, and Y. Rogers. Huddlelamp: Spatially-aware mobile displays for ad-hoc around-the-table

collaboration. In *Proc. ITS '14*, pp. 45–54. ACM, 2014. doi: 10.1145/2669485.2669500

[45] R. Rädle, H.-C. Jetter, M. Schreiner, Z. Lu, H. Reiterer, and Y. Rogers. Spatially-aware or spatially-agnostic?: Elicitation and evaluation of user-defined cross-device interactions. In *Proc. CHI '15*, pp. 3913–3922. ACM, 2015. doi: 10.1145/2702123.2702287

[46] J. Roberts, P. Ritsos, S. Badam, D. Brodbeck, J. Kennedy, and N. Elmqvist. Visualization beyond the desktop–the next big thing. *IEEE Computer Graphics and Applications*, 34(6):26–34, Nov 2014. doi: 10.1109/MCG.2014.82

[47] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. CMV '07*, pp. 61–71. IEEE Computer Society, 2007. doi: 10.1109/CMV.2007.20

[48] R. Sadana and J. Stasko. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proc. AVI '14*, pp. 265–272. ACM, 2014. doi: 10.1145/2598153.2598163

[49] R. Sadana and J. Stasko. Designing multiple coordinated visualizations for tablets. *Computer Graphics Forum*, 35(3):261–270, 2016. doi: 10.1111/cgf.12902

[50] R. Sadana and J. Stasko. Expanding selection for information visualization systems on tablet devices. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, ISS '16, pp. 149–158. ACM, New York, NY, USA, 2016. doi: 10.1145/2992154.2992157

[51] M. Schreiner, R. Rädle, H.-C. Jetter, and H. Reiterer. Connichiwa: A framework for cross-device web applications. In *Proc. CHI EA '15*, pp. 2163–2168. ACM, 2015. doi: 10.1145/2702613.2732909

[52] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, Nov 1994. doi: 10.1109/52.329404

[53] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *IEEE Symp. on Visual Languages*, pp. 336–343, Sep 1996. doi: 10.1109/VL.1996.545307

[54] M. Spindler, M. Schuessler, M. Martsch, and R. Dachselt. Pinch-drag-flick vs. spatial input: Rethinking zoom & pan on mobile displays. In *Proc. CHI '14*, pp. 1113–1122. ACM, 2014. doi: 10.1145/2556288.2557028

[55] M. Spindler, C. Tominski, H. Schumann, and R. Dachselt. Tangible views for information visualization. In *Proc. ITS '10*, pp. 157–166. ACM, 2010. doi: 10.1145/1936652.1936684

[56] M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1065–1072, 2009. doi: 10.1109/TVCG.2009.162

[57] C. Tominski, C. Forsell, and J. Johansson. Interaction support for visual comparison inspired by natural behavior. *IEEE Trans. Vis. Comput. Graphics*, 18(12):2719–2728, Dec 2012. doi: 10.1109/TVCG.2012.237

[58] P. Woźniak, L. Lischke, B. Schmidt, S. Zhao, and M. Fjeld. Thaddeus: A dual device interaction space for exploring information visualisation. In *Proc. NordiCHI '14*, pp. 41–50. ACM, 2014. doi: 10.1145/2639189.2639237

[59] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Vis. Comput. Graphics*, 13(6):1224–1231, Nov 2007. doi: 10.1109/TVCG.2007.70515