Integrating 2D Mouse Emulation with 3D Manipulation for Visualizations on a Multi-Touch Table

Luc Vlaming,[1] *Christopher Collins,*[2] *Mark Hancock,*[3]
Miguel Nacenta,[4] *Tobias Isenberg,*[1,5] *Sheelagh Carpendale*[4]

[1]University of Groningen
{l.vlaming@ | isenberg@cs.} rug.nl

[2]University of Ontario Institute of Technology
christopher.collins@uoit.ca

[3]University of Waterloo
mark.hancock@uwaterloo.ca

[4]University of Calgary
{manacent | sheelagh}@ucalgary.ca

[5]DIGITEO &
CNRS/INRIA

ABSTRACT

We present the Rizzo, a multitouch virtual mouse that has been designed to provide the fine grained interaction for information visualization on a multi-touch table. Our solution enables touch interaction for existing mouse-based visualizations. Previously, this transition to a multi-touch environment was difficult because the mouse emulation of touch surfaces is often insufficient to provide full information visualization functionality. We present a unified design, combining many Rizzos that have been designed not only to provide mouse capabilities but also to act as zoomable lenses that make precise information access feasible. The Rizzos and the information visualizations all exist within a touch-enabled 3D window management system. Our approach permits touch interaction with both the 3D windowing environment as well as with the contents of the individual windows contained therein. We describe an implementation of our technique that augments the VisLink 3D visualization environment to demonstrate how to enable multi-touch capabilities on all visualizations written with the popular prefuse visualization toolkit.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces—Graphical user interfaces, interaction styles.

General terms: Design, Human Factors

Keywords: virtual mouse, multi-touch, information visualization.

INTRODUCTION

With the recent surge of touch technology [9, 13, 25, 30, 32] it is increasingly possible to create multi-touch environments that support collaboration and enable interfaces with rich, direct manipulation. However, there is still relatively little research into practical and collaborative information-intensive applications. These applications have their own requirements for interaction and usually have fine-grained information aspects that require a high input resolution. In a traditional desktop environment these information-rich visualizations would commonly make use of mouse interaction to provide this fine-grained information access. Our challenge is to move these detail-intensive information visualizations into a multi-touch tabletop environment, gaining some of the freedoms and advantages of multi-touch interaction without losing the fine-grained information access.

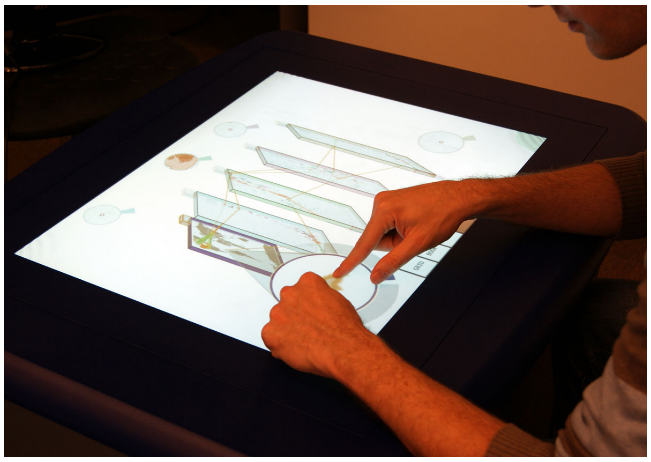
Figure 1: With our system, a person can interact with information visualizations using the Rizzo, a tool designed to enable mouse interaction.

To create an environment that functions coherently as a multi-touch interaction space and at the same time provides detailed and precise information access, we designed a tool, called the Rizzo (Fig. 1), which is a virtual mouse that translates the multi-touch interactions on the surface to mouse input on the visualizations. To explore this design challenge we chose to work with an existing information visualization system, VisLink [7], that provides an interactive, comparative visualization environment, which can hold any number of visualizations on independent panels in a 3D space. Because we work with VisLink, we enable the use of any infor-

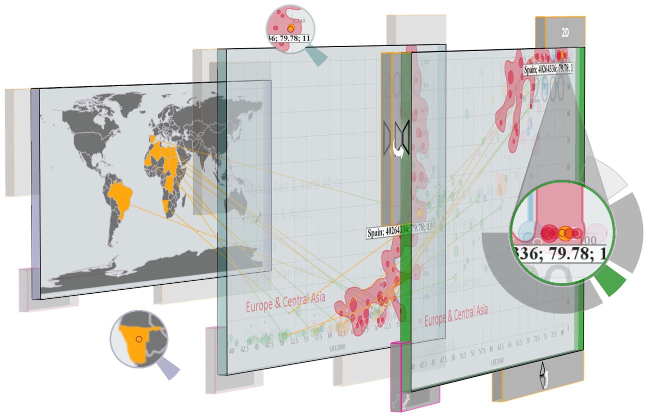

Figure 2: Rizzos provide resolution-aware mouse emulation for 2D visualizations embedded on panels in the 3D environment.

mation visualizations written with the widely used prefuse toolkit [17]. These visualizations are integrated in our system through panels within VisLink's 3D space, a setup that places high demands on the design of the Rizzo, since our tool needs to be general enough to work with any visualization, and simultaneously enable effective multi-touch manipulation. We explore this design space of mixed 2D and 3D interaction that allows people to interact in a unified multi-touch interaction continuum for information-intensive tabletop applications.

Our main contribution is the Rizzo (Fig. 2); a tool that enables precise selection and manipulation of information elements within the 2D panels using multi-touch input. We also contribute a group of interaction techniques that allow the 3D manipulation of information visualization panels. This includes controlling the 3D view and the 3D layout and being able switch to and from a 2D-only view.

We first explain our design decisions and describe the integration of 3D navigation, 3D manipulation of the panels, and 2D visualization interaction within the panels. We then discuss how our system would be used in practice, provide an informal evaluation, and also discuss the benefits and limitations of our approach.

RELATED WORK

Our research builds upon three main areas of interest: multi-touch manipulation on interactive surfaces; integrated 2D/3D window management systems; and precise interaction and mouse emulation on multi-touch surfaces.

Multi-Touch Manipulation

The introduction of multi-touch technology on digital surfaces [9, 13, 25, 30, 32, 35] has led to a variety of techniques for manipulating virtual artifacts with one's fingers. Many techniques have since been introduced for moving, rotating, and scaling 2D artifacts with one [21, 31] and several fingers [16]. These general ideas have been extended to the manipulation of 3D objects on touch surfaces with two or more fingers [14, 15, 29, 39]. As an alternative to this, it is possible to use simulated physics to control objects [37, 38].

While these techniques allow us to manipulate and transform virtual 2D and 3D artifacts on a multi-touch surface, they do not discuss how to interact with the information contained within the artifact (*e. g.*, select a word, copy & paste). In our approach we integrate 3D multi-touch manipulation with the ability to interact directly with the information contained by these artifacts. Hence, we use both the sticky fingers and opposable thumbs technique [15] and precise constrained manipulations, as well as mouse-like control of panels containing visualizations.

3D Window Management Systems and Multi-Touch

The placing of objects containing information into a 3D environment as done in VisLink [7] is reminiscent of 3D window management systems including Miramar [23], Metisse [6], BumpTop [2], and Caleydo [22, 36]. Bringing these hybrid 2D-within-3D window management systems to a multi-touch interaction setup (*e. g.*, a multi-touch tabletop display) promises to enhance the experience over current mouse-based interactions. Multi-touch input offers more degrees of freedom, therefore providing greater flexibility in the design of interaction techniques. Furthermore, tabletop displays promote collaboration, an important aspect of many scenarios, including information visualization analysis [18]. The traditional ways of interacting with 3D window management systems and VisLink [7], however, are mouse- and keyboard-based, so switching to multi-touch interaction introduces issues such as precision and dedicated mouse mappings when interacting with window content. In fact, previous attempts to add multi-touch interaction to 3D window management systems (*e. g.*, BumpTop) only offer mappings for very specific information types. We aim to combine the advantages and generality of multi-touch and mouse interaction.

Precise Tabletop Interaction and Mouse Emulation

Previous studies comparing indirect mouse to direct-touch input [12] showed that a mouse may be more appropriate for unimanual input tasks, and fingers for a bimanual input tasks. However, on digital tabletop settings it is often awkward to provide additional room for mice, while combined relative and absolute pointing techniques [11] using pen input cannot make use of multi-touch capabilities. Thus, we explore a setting that allows us to make use of both direct-touch manipulations of 3D objects and indirect mouse emulation.

One of the advantages of mouse interaction over touch input is its high precision. This issue has been addressed by several techniques such as virtual key tapping, a gain-based cursor [3], stretching methods, menus for parameter setting [5], localized views [34], and a rubbing gesture for zoom control [28]. We integrate high-precision control into our mouse emulation via a lens with controllable zoom and a non-linear control-display gain.

Several other techniques exist that enable mouse interaction on multi-touch displays. Esenther *et al.* [10] presented the Fluid DTMouse, an interaction technique in which the number of touch points defines the mouse operation to apply. Matejka *et al.* [24] improved on the Fluid DTMouse and recommend SDMouse, a set of techniques to enable three-button mouse capabilities through touch gestures. We build on these techniques but realize mouse emulation through a dedicated tool rather than through gestures. In that way our mouse emulation is related to Microsoft's virtual two-button mouse [26]. However, we integrate lens functionality, absolute and relative pointing, and precision control for information exploration. Also, our tool was designed to operate in a holistic setting which simultaneously provides multi-touch support in the 3D setting and 2D interaction with displayed information. We know of no other techniques which specifically address the situation of providing mixed 2D and 3D interaction. Indeed, many of the multi-touch gestures used for mouse emulation overlap those for 3D manipulation, so the reported techniques cannot be used together in our system.

SETTING THE STAGE FOR THE RIZZO

The overarching goal of our work is to create an integrated tabletop information analysis environment, where multi-touch capabilities can help small teams of information workers tackle their increasingly complex tasks. This is a large goal comprised of many components that will fuel considerable research. For example, in establishing a collaborative interaction environment, many interaction factors have been identified as being important, including interaction awareness cues, simultaneous interaction, and support of changing collaboration styles. Also, a flexible workspace organization has been shown to be important for comprehension—readability of 2D information can be improved through changing orientation; communication—repositioning and resizing of information can support information sharing; and coordination—repositioning of information items allows people to adjust collaborative working styles such as joint and parallel work [20]. While these factors, particularly the flexibility of positioning 2D information, influenced our choice of environment, our main focus in this current research is to design and develop a virtual mouse that can touch-enable the interactions previously provided by a mouse for a set of visualizations. We thus set the stage for our work on the Rizzo by choosing a set of visualizations and a multi-touch interaction approach.

Use of VisLink. In order to concentrate on the development of the integrated multi-touch interactions, we chose to use an existing visualization platform, VisLink [7]. VisLink extends prefuse [17] by creating a linked multiple panel environment. VisLink is a visualization platform in which multiple 2D visualizations can be imported onto 2D panels in a 3D environment. VisLink supports directly importing visualizations created within prefuse for 2D, preserving their original 2D interactions. This concurs with our intentions of grandfathering the visualization's interactions within our environment and provides some generality because any visualization written in prefuse can be imported into VisLink, thus also into our environment. The use of VisLink provides us with: a suite of existing 2D visualizations that can be imported as is, each in their own panel; a 3D environment which holds the 2D visualization panels; and preserved internal visualization interactions that work according to the original visualizations. In addition, VisLink offers the potential for creating queries and visualizing links between two or more of the 2D visualizations according to selected semantics. Thus, VisLink provides many of our visualization requirements and allows us to concentrate on the challenges of creating a virtual mouse in a mixed 2D and 3D interaction environment.

Use of Sticky Tools A variety of methods for manipulating 3D virtual objects on multi-touch surfaces have recently been introduced [14, 15, 29, 38, 39]. For our purposes, we chose Sticky Tools [15], which provide direct multi-touch interaction that offers full 6DOF control. We explore the possibility of using these 3D manipulation techniques for interacting with 2D panels that contain the information visualizations. People can, *e. g.*, move, rotate, and otherwise arrange the visualization panels to suit the needs of their information exploration.

DESIGN CHALLENGES

Although we leverage previous work (VisLink and the Sticky Tools concept), creating the Rizzo still poses many design challenges. For example, how can such technique enable a the use of familiar information tools? how can it enable a flexible organization and manipulation of the space? how can it allow collaboration? In this section, we describe the primary design challenges we encountered throughout our design process.

Emulate Mouse Interactions. Within VisLink, most of the visualization techniques are currently mouse-based. Thus, providing mouse-like interaction with these 2D information visualizations will be necessary to enable each visualization's unique interaction techniques. For example, some visualizations may support zooming while others may not, and zooming may have different meanings within the context of different visualizations. Since our intention is to support importing of arbitrary legacy visualizations, our solution must be independent of internal legacy visualization choices. Thus, in our design we have chosen to provide a method of invoking standard mouse operations, which can be interpreted by all visualizations in the prefuse toolkit. There have been a variety of different methods introduced for emulating mouse interaction on multi-touch tables [10, 24]. However, we want to develop a virtual mouse that will, if possible, alleviate visualization interaction issues in precise, possibly even subpixel, selections—or at the very least not exacerbate this. We also want to provide a harmonious interaction set where a person can do simple finger and hand interactions to perform simple activities.

Combine Precise Selection & Direct Manipulation. While multi-touch technology promises a direct connection between the people and the information they are controlling, the use of hands and fingers can result in the loss of precision. When analyzing data, it may be necessary to precisely select and manipulate information to gain insight or verify hypotheses about specific elements of the data. Being able to directly touch the data to select and manipulate may more closely match a person's expectations. Thus, our mouse emulation should ideally support both precise selection and direct manipulation.

Support Simultaneous Interaction. Most existing information visualizations support interaction using only one mouse. Many potential conflicts have been identified when trying to use multiple mice to control an application designed for only one mouse [33]. Consequently, enabling multiple mice in each visualization panel may lead to these same con-

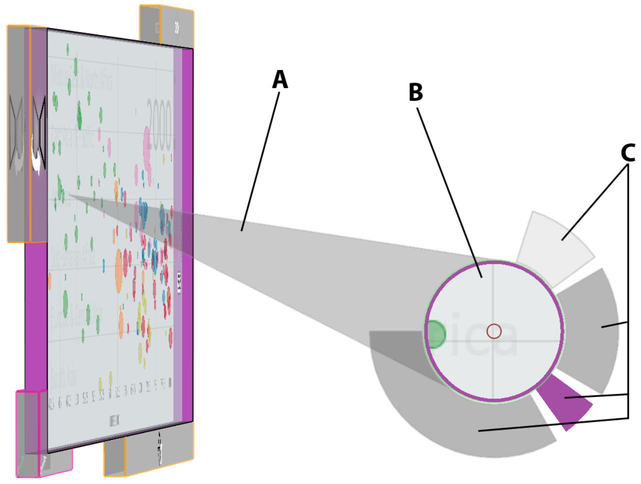

Figure 3: The basic anatomy of the Rizzo. A is the cone, B the base, and C are the wedges.

flicts. Providing each visualization panel with its own virtual mouse, avoids this potential conflict.

In the following sections we describe the specific interaction techniques that comprise our information analysis environment.

RIZZO, A MULTI-TOUCH VIRTUAL MOUSE

The central component of our system is the Rizzo, a multi-touch virtual mouse that allows us to interact with information contained in the 2D panels. The Rizzo was designed to mediate effective interaction with visualization components, and therefore we explicitly set out to achieve a technique that: a) allows passing the same events that legacy applications require (*i. e.*, regular mouse events), b) makes good use of the interaction and ergonomic advantages of multi-touch input, c) is as unobtrusive as possible to avoid interference with the main goal of the system (visualization analysis), and d) enables precise interaction with any size of element within any visualization. Each Rizzo instance is associated with a specific visualization component.

Rizzo's Anatomy

The Rizzo can be divided into three main interaction areas (see Fig. 3): the cone (A), the base (B), and the wedges (C). The cone of Rizzo is a translucent conical shape that connects the cursor tip (the point where the actions of the virtual mouse take place) with a circular area called the base (B). The center of the base holds the lens, a representation of the area of the visualization currently surrounding the cursor tip. The lens takes the majority of the space of the base, except for the base's rim, which provides an association with its visualization component through its color. The wedges (C) are pie-shaped soft buttons located radially around the center of the base which serve to emulate the button clicks of a mouse and to configure the parameters of its operation. One of these wedges is called the color code wedge (see B in Fig. 4), and it acts as a handle for the base.

The size of the base can also be altered by pinching on any two points of the base, which enlarges the area occupied by

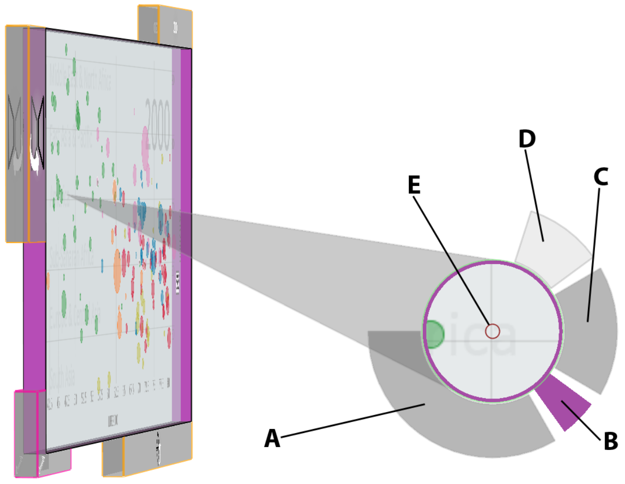

Figure 4: The wedges of the Rizzo. A and C provide respectively left and right mouse button functionality, B is a colored wedge to allow usage of the Rizzo without occluding the lens. D is used to control the zoomlevel of the lens. E is a red circle that provides an indication of where the actual mouse cursor is on the visualization.

the lens, but not the region of the visualization that it covers (i.e., enlarging the lens implicitly increases the zoom).

Looking through the Rizzo

The base of the Rizzo provides an undistorted view of the area of the visualization surrounding the cursor point. The Rizzo's lens acts as a visualization lens to its target visualization region (providing a possibly zoomed-in clip of the cursor region) and as a thumbnail; it adds a visual link between the visualization and its Rizzo. The level of zoom can be controlled through one of its wedges (see D in Fig. 4) by touching it and describing circles around the base, similar to zooming a DSLR lens. The size of the base can also be altered by pinching on any two points of the base, which enlarges the lens and thus increases the visualization zoom level.

Note that the representation of the visualization on the lens is always parallel to the surface of the display, regardless of the position and orientation of the 3D panel containing the data. In other words, the lens representation billboards the view of the visualization content, the latter quite often being skewed in perspective. It is also important to note that, because the visualizations can be moved in 3D space, the contents of some of the visualization panels can be occluded by other panels or interface elements; the Rizzo lens helps address this issue because the Rizzo is always represented on top of all other objects, and the Rizzo lens can display content that is otherwise occluded.

Moving the Rizzo

The Rizzo needs to provide flexible ways to move the mouse cursor as well as the Rizzo itself. Simultaneously, it is necessary to provide sufficient input resolution to enable the manipulation of the smallest elements in the visualization and to provide quick access to all parts of the visualization areas. This is achieved in the Rizzo through three ways of

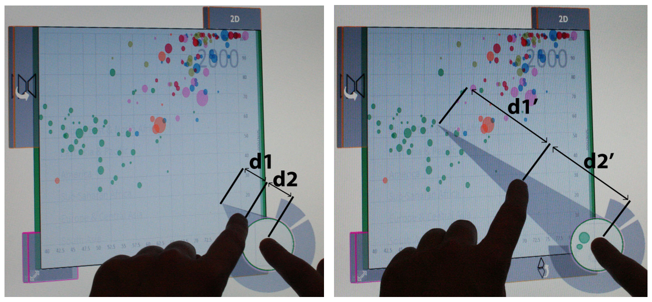

Figure 5: Pantograph pointing with the Rizzo. The relationship of the distances between the cursor and cone touch ($d1$) and the cone touch and lens touch ($d2$) is preserved: $\frac{d1}{d2} = const = \frac{d1'}{d2'}$.

changing the position of the cursor, or the cursor and the base.

Relative Pointing. Touch-dragging any point of the Rizzo base attaches the base to the finger (as when dragging an object through direct-touch), but also changes the cursor position within the visualization pane in the same direction as the movement of the touch-drag (relative to the table). Although the direction of movement of the finger and the cursor are parallel, the control-display gain is not linear. This means that slow movements of the finger will produce shorter displacements of the cursor per distance traveled than fast movements. This is an indirect relative mapping analog to the cursor *acceleration* used in most operating systems, and has the purpose of facilitating quick access to distant areas of the visualization without having to sacrifice pointing resolution.

If the movement of the finger makes the cursor pointer reach the boundary of the visualization pane, the lens will keep moving with the finger, but the cursor point will stay within the visualization (analogous to the cursor's behavior in most operating systems when the cursor reaches the end of the screen).

Offset Pointing. If a touch-drag is started within the narrowest third of the tip of the cone while another touch is activated on the lens area, the cursor will move with a constant offset from the cone-touch. This behavior is equivalent to the offset cursor in [4], except for the lens-touch, which is required to keep the Rizzo active (if the lens is not touched, the Rizzo's cone is transparent to interaction to allow direct interaction with the visualization objects). This type of pointing is absolute and indirect, and is useful when the 1-to-1 relationship between input and display needs to be preserved, but the finger needs to be out of the way. Once the cursor cone is being touched, the first touch on the lens can be released.

Pantograph Pointing. If a touch-drag starts on the widest two-thirds area of the cone while another touch is active on the lens area, the cursor will move to preserve the relationship of the distances between the cursor and the cone-touch ($d1$) and the cone-touch and the lens-touch ($d2$), *i. e.*, $\frac{d1}{d2} = const$ (see Fig. 5). This is equivalent to the pantograph technique [8] but with two touches (similar to the basic movement of the two-handed technique presented in [1]). This way of moving the cursor allows people to interact comfortably from a distance. Once the cursor cone is touched, the first touch on the lens can be released.

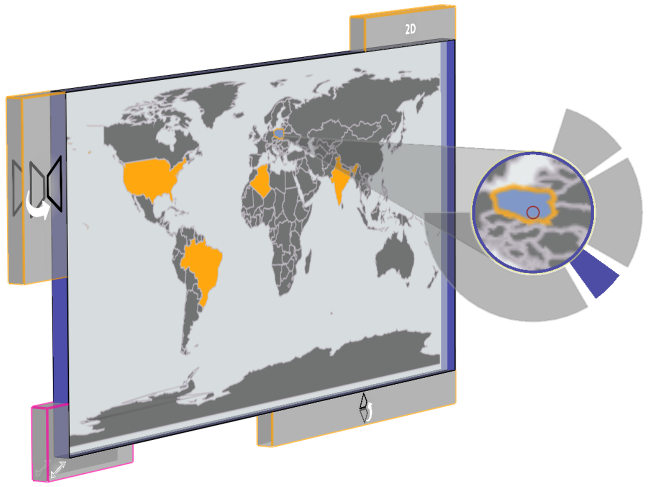

Figure 6: The Rizzo being used on a map panel.

Acting with the Rizzo

The Rizzo also allows clicking and dragging (part A and C, see Fig. 4) through the wedge buttons around the lens area. These buttons emulate mouse buttons, including their state (for dragging), and the fact that they stay in the same orientation with respect to the table to facilitate vision-less operation by experts. The exact location pointed at by the cone tip is represented within the lens by a circle (part E in Fig. 4) to facilitate clicking from the lens without having to look at the tip of the cone.

Resting the Rizzo

To avoid clutter and occlusion, all the elements of Rizzo, except the base and the color code wedge, fade away within a few seconds when not in use. If the Rizzo is not activated, its lens holds its position with respect to the table, and its cone tip (the cursor) holds its position with respect to its visualization panel, even if the panel is moved.

Using the Rizzo

The features described above correspond to our design goals. The Rizzos generate the same events as regular mice, but also add extra functionality. For example, several Rizzos can simultaneously manipulate different visualization panels (taking advantage of multi-touch interaction, the Rizzos avoid occlusion by fading away when not needed (respecting the main goal: visualization), they enable different ways of moving the cursor (to provide adequate resolution and comfortable reach), and the Rizzos also provide on-demand magnification of visualization regions (visual resolution).

3D NAVIGATION

We also require 3D navigation of the 3D virtual world, where the use of touch is in harmony with the touches used with the Rizzo. Following the design of the original VisLink environment, we support navigation through interactions designed to change the view. Difficulties with such camera manipulations are common sources of usability headaches when interacting with 3D visualizations. For example, poor design

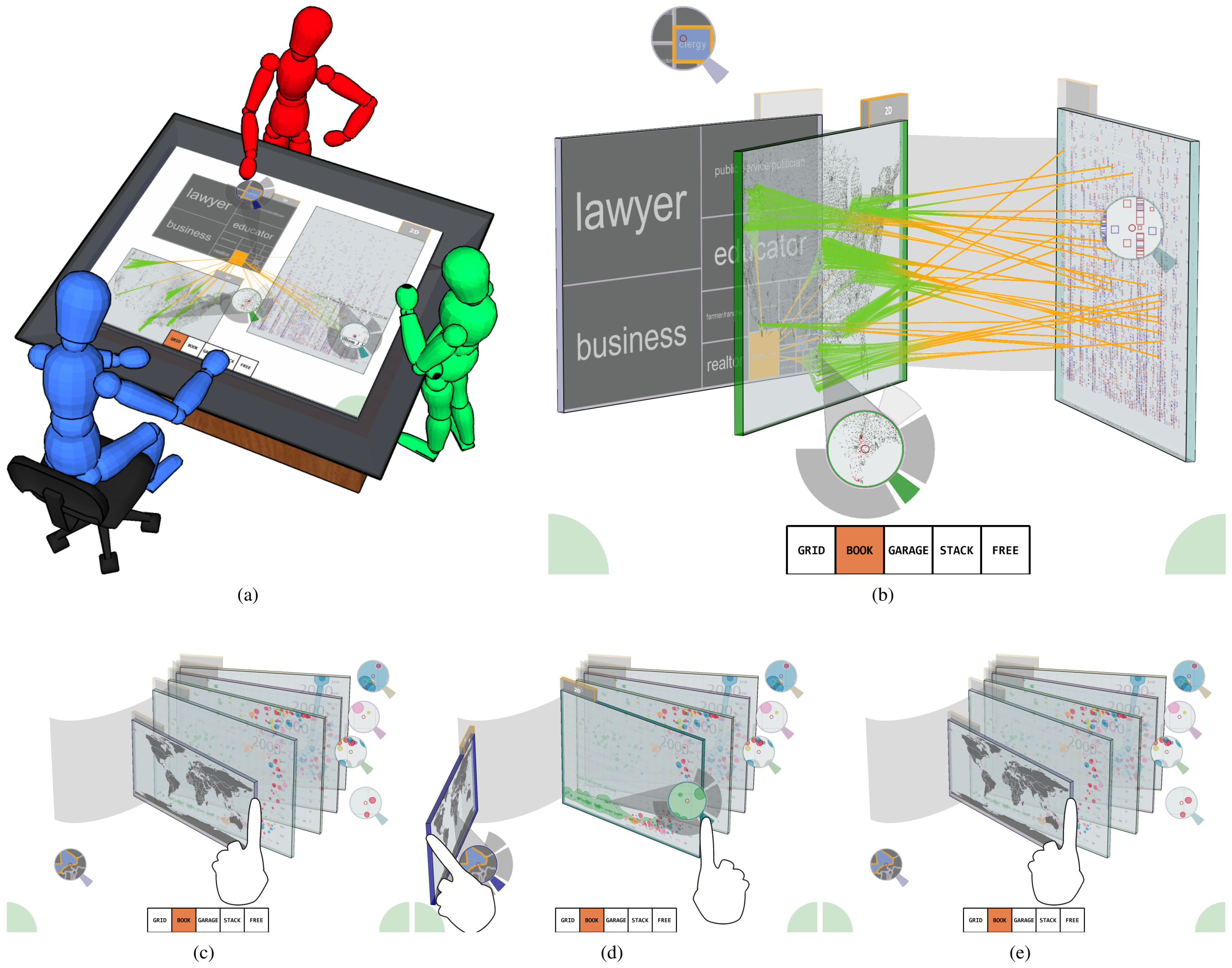

Figure 7: This series of images shows an example of how our system might be used by analysts. An expected scenario would involve many people gathering to analyze information (a). An analyst can use the VisLink system to view connections between two visualizations (b). A more complex interaction made possible by our system might involve an analyst first grabbing a map panel (c), and then dragging it to the left so that she can use the Rizzo on the bubbleset panel (d), and then dragging the map back (e).

can lead to analysts becoming lost when confronted by empty space without any visual anchors that would support reorientation. To avoid this pitfall, we fix the camera's viewing target to always point to the centre of the 3D space upon which the panels are initially positioned. We use a virtual trackball to manipulate the camera position: touch-and-drag operations on the empty background areas cause movements of the camera across the surface of a sphere such that the scene contents appear to rotate in unison. Pinching gestures on the background are used to zoom in or out, which actually manipulates the distance from the scene centre to the camera position. In order to ensure that navigation will always be possible, even when no empty background areas are visible, we added two always-on-top navigation areas in the corners of the screen (*e. g.*, Fig. 7(b)).

3D OBJECT MANIPULATION

Within the 3D environment, we have the 2D panels that hold the visualizations. Interaction is required to manipulate the panels in the 3D space. Here we provide two types of control: free 6DOF interaction and several constrained strategies. For the unconstrained 6DOF control, we use the Sticky Tools technique [15], which we selected over other techniques due to the simplicity of its implementation and the control it provides. Familiar multi-touch translation and resize operations, such as two-finger pinching (simultaneous resize and translate) and one-finger drag (translation), are gestures that can be performed anywhere on a panel. When using the Sticky Tools technique in a test session, however, we quickly concluded that this unconstrained manipulation technique introduces problems for analytical comparisons. For example, making comparisons between two visualizations by aligning them parallel, or side by side, is difficult. It has

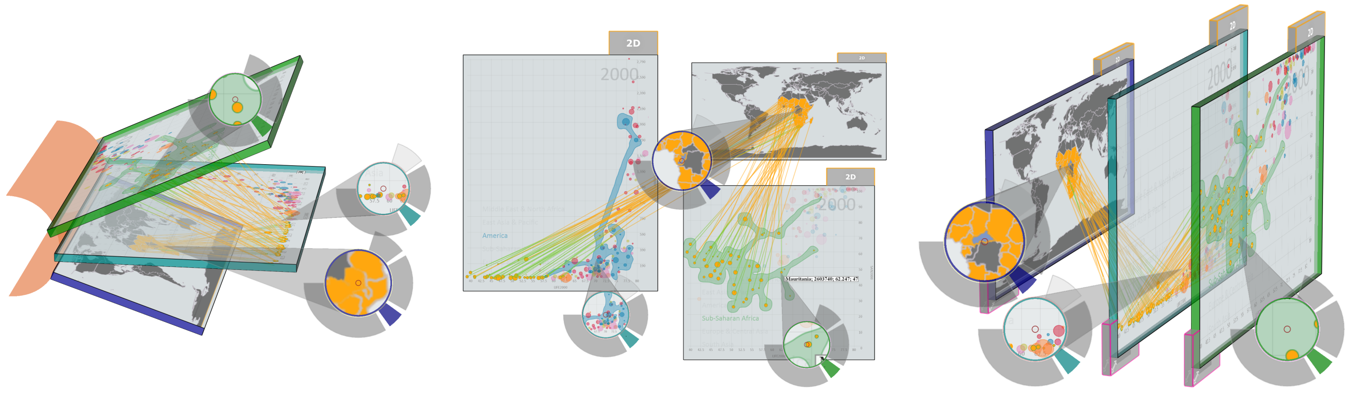
Figure 8: Constrained comparison modes: garage door opening (book opening is equivalent), grid, and stack.

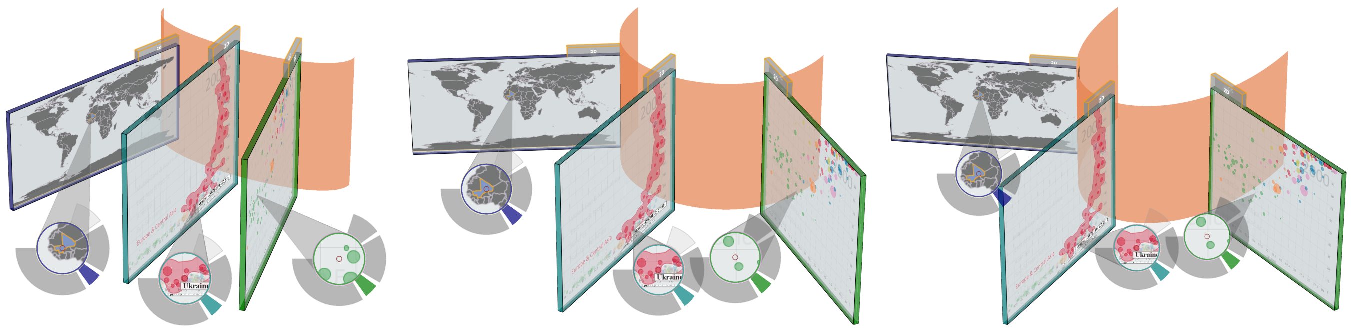
Figure 9: The path the visualization panels follow can be influenced by bending it.

been shown that precisely aligning two objects in 2D, while having control over both translation and rotation, is difficult [27]. When a third dimension is added, using all 6DOF to move objects compounds the precise alignment problem.

Since, as just noted, using unconstrained Sticky tools techniques introduced problems for precise positioning, we employed several types of constrained manipulation interactions without affecting the 6DOF unconstrained manipulation. First, rotation around the *x*- and *y*-axes is enabled with 'rotation' buttons at two sides (see Fig. 6). These buttons are large enough to accommodate the lower input precision on multi-touch devices. Dragging a button causes rotation about the axis along the opposite side of the panel. In addition, resizing is enabled with a button at one corner. Since, when a panel can be oriented such that its narrow edge is presented to the view, the acquisition of both the panel's buttons and the panels as a whole is facilitated by widening the panel frame, increasing the touch acquisition area.

In addition, we provide several analytically preferable views, along with interaction techniques for constrained movement of the panels, while maintaining the chosen analytic views. The provided views are *garage door*, *book*, *grid*, and *stack* and are shown in Figures 8 and 9. The preferred views of 'garage door' and 'book' were augmented with a dedicated path along which the panels can move. This path acts as an anchor connecting the panels—as a spine would connect the pages of a book. Dragging with a single finger on the panel moves the panel along the path, while the curvature of the path can be adjusted by a one-finger touch and drag operation (see Fig. 9). The grid arranges the panels adjacent to each other to permit side-by-side analysis. The stack, in contrast, allows comparison between adjacent panels by aligning them in parallel. Translation in the stack orientation occurs along a straight line connecting the panel centers.

Finally, to provide a traditional view on each visualization, we also offer the option to show each visualization in 2D by itself. Switching back and forth between this 2D mode and the previous view is achieved by tapping the '2D' button at the top of each panel.

Together, these techniques and views provide the analyst with options to manage the visualization panel configuration either freely or in a controlled manner to place panels into a desired arrangement for analysis.

IMPLEMENTATION CONSIDERATIONS

As our implementation was designed to support the use of legacy applications in a multi-touch environment, it is written in Java atop the prefuse visualization toolkit [17] and the VisLink visualization platform [7]. VisLink's controller was augmented to handle the multi-touch events corresponding to window and view management in 3D, and to provide the virtual mice which pass 2D events to the constituent visualizations for handling. Existing, stand-alone prefuse visualizations, thus, only have to be adjusted slightly to be integrated, equivalently to their adjustment for the original VisLink system. In fact, the input controller within VisLink behaves similarly to any window management system—input device events are passed to the underlying application (visualization) and handled at that level. This means that visualizations handle mouse events as appropriate without requiring re-engineering of the input handling within the individual visualization's source code. While this may mean that the same mouse events have different results depending on the visualization panel receiving the event, it also means that people familiar with the interaction conventions of a particular visualization can use it with Rizzo without re-learning how to use each visualization.

Our prototype can receive inputs from several multi-touch hardware devices, including the Microsoft Surface, the DViT, and the SMART Table. Input events are read from an XML stream provided by the input device driver. This means that adding support for additional devices such as TUIO-compliant touch surfaces [19] only requires capturing the events and sending them as an XML stream on the local host.

INFORMAL EVALUATION

To provide some initial feedback, we provided a demonstration of our prototype to a class of undergraduate human-computer interaction students. While these students do not represent our expected domain experts, their feedback led to several interesting observations. First, the students did not perceive the Rizzo to be providing the functionality of a mouse, but instead saw it as a lens through which they could examine the data. Despite not recognizing it as a virtual mouse, they had no difficulty making use of the basic mouse functionality, such as moving the cursor and selecting data. Particularly surprising was the near immediate ability to use pantograph pointing. Without being explicitly taught about the relationships which define the cone length, students were able to make use of this technique to perform selections. In fact, we observed several students who used the pantograph pointing exclusively, rather than in conjunction with precise offset pointing through moving the Rizzo base. This preference for using the cone resulted in a reduced ability to successfully select small items. This may be due to the minimal instructions that were given, or it may indicate a problem with our precise pointing technique or visual design.

When exploring our United Nations dataset, students found interesting comparisons, such as the healthcare spending differences amongst their countries of origin. While each default view was explored and revisited, the book view seemed to be preferred when comparing scatterplot visualizations. The grid view was used as an overview of all visualizations, in order to select which visualization to focus on next.

The students also did not make use of the system in a multi-user fashion, and instead tried the demo out one at a time. They may have chosen this turn-based approach due to the physical size of the table (the SMART Table was initially designed for use by children), but this may also be due to our design decisions involving the visual layout of the interface. Specifically, the controls to switch between different views imply a preferred side of the table. For example, it may be more appropriate to replicate these controls, or to use orientation-independent icons in the display's corners.

DISCUSSION AND CONCLUSION

In this paper we have presented an approach that enables both high-level multi-touch interaction with 3D objects that carry information visualizations and fine-grained control of the visualizations themselves via the Rizzos. The design of the interaction techniques was driven by the goal to use mouse-based prefuse visualizations in a unified interaction space that is controlled using multi-touch input. This allows us not only to use existing visualizations with their specific interaction mappings on multi-touch screens, but also to compare and relate them to each other.

The design of the Rizzo was created so that it does not interfere with the multi-touch control of the objects carrying the information visualization while at the same time providing flexible and high-precision mouse control as known from physical mice. This is realized by integrating tool-glass functionality into the mouse, which allows us to zoom and position the mouse pointer precisely. Also, the Rizzo can easily be resized to adjust it to people's hand sizes, such as hands of children or adults.

Our design also leaves a number of points for discussion. For example, our mouse only supports two buttons, relative and absolute adjustment of the pointer, and zooming of the tool-glass functionality. A virtual scroll-wheel, however, was not included in the realization for several reasons. In addition to the added complexity, we also saw issues with the missing feedback for the individual steps of a physical wheel action, turning the scroll-wheel being a discrete action as opposed to the continuous sliding along the touch surface. Another issue is that, in particular in multi-user scenarios, a fluid switching between right-handed and left-handed mouse designs would be desired which is currently not supported.

In general, more research needs to be done to explore the impact of the presence of virtual mice in a multi-user environment. Similarly, the use of virtual mice may differ between small and large surfaces as well as between table and wall environments. Also, currently only one mouse is supported per visualization panel.

ACKNOWLEDGEMENTS

We would like to thank Natural Science and Engineering Research Council of Canada, Alberta's Informatics Circle of Research Excellence, Alberta Ingenuity, and the Canadian Foundation of Innovation for research support. We also thank the reviewers and members of the iLab for their helpful comments on this work.

REFERENCES

1. M. Abednego, J.-H. Lee, W. Moon, and J.-H. Park. I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *Proc. ITS*, pp. 61–64. ACM, 2009. doi: 10.1145/1731903.1731917

2. A. Agarawala and R. Balakrishnan. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI*, pp. 1283–1292. ACM, 2006. doi: 10.1145/1124772.1124965

3. P.-A. Albinsson and S. Zhai. High precision touch screen interaction. In *Proc. CHI*, pp. 105–112. ACM, 2003. doi: 10.1145/642611.642631

4. P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-Pop and Drag-and-Pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. IFIP World Computer Congress*, pp. 57–64, 2003.

5. H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *Proc. CHI*, pp. 1263–1272. ACM, 2006. doi: 10.1145/1124772.1124963

6. O. Chapulis and N. Roussel. Metisse is not a 3D desktop! In *Proc. UIST*, pp. 13–22. ACM, 2005. doi: 10.1145/1095034.1095038

7. C. Collins and S. Carpendale. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1192–1199, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70611

8. M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving drag-and-drop on wall-size displays. In *Proc. Graphics Interface*, pp. 25–32. Canadian Human-Computer Communications Society, 2005.

9. P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *Proc. UIST*, pp. 219–226. ACM, 2001. doi: 10.1145/502348.502389

10. A. Esenther and K. Ryall. Fluid DTMouse: Better mouse support for touch-based interactions. In *Proc. AVI*, pp. 112–115. ACM, 2006. doi: 10.1145/1133265.1133289

11. C. Forlines, D. Vogel, and R. Balakrishnan. Hybrid-Pointing: Fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST*, pp. 211–220. ACM, 2006. doi: 10.1145/1166253.1166286

12. C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI*, pp. 647–656. ACM, 2007. doi: 10.1145/1240624.1240726

13. J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST*, pp. 115–118. ACM, 2005. doi: 10.1145/1095034.1095054

14. M. Hancock, S. Carpendale, and A. Cockburn. Shallow-depth 3D interaction: Design and evaluation of One-, two- and three-touch techniques. In *Proc. CHI*, pp. 1147–1156. ACM, 2007. doi: 10.1145/1240624.1240798

15. M. Hancock, T. ten Cate, and S. Carpendale. Sticky tools: full 6DOF force-based interaction for multi-touch tables. In *Proc. ITS*, pp. 133–140. ACM, 2009. doi: 10.1145/1731903.1731930

16. M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen. Rotation and translation mechanisms for tabletop interaction. In *Proc. TABLETOP*, pp. 79–88. IEEE Computer Society, 2006. doi: 10.1109/TABLETOP.2006.26

17. J. Heer, S. K. Card, and J. A. Landay. prefuse: A toolkit for interactive information visualization. In *Proc. CHI*, pp. 421–430. ACM, 2005. doi: 10.1145/1054972.1055031

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

19. M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. TUIO – A protocol for table-top tangible user interfaces. In *Proc. GW*, 2005.

20. R. Kruger, S. Carpendale, S. D. Scott, and S. Greenberg. How people use orientation on tables: comprehension, coordination and communication. In *Proc. GROUP*, pp. 369–378. ACM, 2003. doi: 10.1145/958160.958219

21. R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proc. CHI*, pp. 601–610. ACM, 2005. doi: 10.1145/1054972.1055055

22. A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context. In *Proc. PacificVis*, pp. 57–64. IEEE Computer Society, 2010. doi: 10.1109/PACIFICVIS.2010.5429609

23. J. Light and J. Miller. Miramar: A 3D workplace. In *Proc. IPCC*, pp. 271–282. IEEE, 2002. doi: 10.1109/IPCC.2002.1049110

24. J. Matejka, T. Grossman, J. Lo, and G. Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. In *Proc. CHI*, pp. 1073–1082. ACM, 2009. doi: 10.1145/1518701.1518865

25. Microsoft. Microsoft Surface. Webpage, http://www.surface.com/, visited Oct. 1, 2010.

26. Microsoft. What is the touch pointer? Webpage, http://windows.microsoft.com/en-US/windows-vista/What-is-the-touch-pointer, visited Oct. 1, 2010.

27. M. A. Nacenta, P. Baudisch, H. Benko, and A. Wilson. Separability of spatial manipulations in multi-touch interfaces. In *Proc. Graphics Interface*, pp. 175–182. Canadian Information Processing Society, 2009.

28. A. Olwal, S. Feiner, and S. Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proc. CHI*, pp. 295–304. ACM, 2008. doi: 10.1145/1357054.1357105

29. J. L. Reisman, P. L. Davidson, and J. Y. Han. A screen-space formulation for 2D and 3D direct manipulation. In *Proc. UIST*, pp. 69–78. ACM, 2009. doi: 10.1145/1622176.1622190

30. I. Rosenberg and K. Perlin. The UnMousePad: An interpolating multi-touch force-sensing input pad. In *Proc. SIGGRAPH*, pp. 1–9. ACM, 2009. doi: 10.1145/1576246.1531371

31. C. Shen, F. D. Vernier, C. Forlines, and M. Ringel. DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proc. CHI*, pp. 167–174. ACM, 2004. doi: 10.1145/985692.985714

32. SMART Technologies. SMART Table interactive learning center. Webpage, http://smarttech.com/table, visited Oct. 1, 2010.

33. E. Tse and S. Greenberg. Rapidly prototyping single display groupware through the SDGToolkit. In *Proc. AUIC*. Australian Computer Society, 2004.

34. D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *Proc. CHI*, pp. 657–666. ACM, 2007. doi: 10.1145/1240624.1240727

35. W. Buxton. Multi-Touch Systems That I Have Known and Loved. Webpage, http://www.billbuxton.com/multitouchOverview.html, visited Oct. 1, 2010.

36. M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proc. Graphics Interface*, pp. 129–136. Canadian Information Processing Society, 2010.

37. A. D. Wilson. Simulating grasping behavior on an imaging interactive surface. In *Proc. ITS*, pp. 125–132. ACM, 2009. doi: 10.1145/1731903.1731929

38. A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing physics to the surface. In *Proc. UIST*, pp. 67–76. ACM, 2008. doi: 10.1145/1449715.1449728

39. L. Yu, P. Svetachov, P. Isenberg, M. H. Everts, and T. Isenberg. FI3D: Direct-touch interaction for the exploration of 3D scientific visualization spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), Nov./Dec. 2010. To appear.