View and Space Management on Large Displays

Anastasia Bezerianos and Ravin Balakrishnan
University of Toronto

The canvas portal framework facilitates the development of alternative view-based space-management techniques on large displays. Implementations of the framework provide layout management, context switching, and space creation.

The research community and design industry have long been interested in interaction with large-format displays, with much of the early research focusing on single whiteboard-sized displays. More recently, the rapidly decreasing cost of projectors has spurred construction of wall-sized displays by tiling multiple projectors to form a single virtual image. These multiprojector displays are particularly interesting from an interaction perspective in that the high resolution provided by the tiling of multiple projectors lets users view high-quality imagery even when they're up close to the display, as in Figure 1. Single-projector systems at that scale present images that are too pixelated for up close interaction.

If we're to use these large-scale displays up close in the highly interactive manner for which they're well suited, we must address the interaction challenges they introduce. Over the past decade, researchers have developed various interaction techniques to address these challenges. However, many of these techniques have been developed and presented in isolation, with no effort to consolidate the solutions into a single unifying framework. We introduce the *canvas portals* framework to address this lack of integration and to act as a catalyst for the development of new techniques.

In the most abstract sense, canvas portals are alternative views of display canvas areas where interacting with the portal's interior is equivalent to interacting with the depicted display area. However, manipulating the various canvas portal parameters (including the mappings between the canvas portals and the depicted area of the underlying display) lets us support a variety of novel interactions. To illustrate the generality of the canvas portals concept, we've instantiated a representative sample of existing large-display interaction techniques within the canvas portal design space, and developed several interaction techniques that address some of the challenges in new ways.

Large-display interaction challenges

A number of interaction challenges arise from the ability of large-scale displays to present vast quantities of data over a large spatial canvas.

Remote reaching

Unlike interaction on a desktop, or even a small whiteboard-sized display, where users can reach most displayed items, data on wall-sized displays often resides in an unreachable location. From a visualization perspective, a user might not be able to view all parts of the screen with equal clarity because some of the display will appear in the user's peripheral vision. As a result, existing user interfaces mapped onto displays of this scale for up close interaction would at the very least require the user to walk around the display to accomplish even simple tasks. These interface elements might be unusable altogether when, for example, the user can't reach the top of the display to operate an application's menu bar. Admittedly, a user could operate such a display from afar, using a mouse and a keyboard, but this approach doesn't fully leverage the potential benefits of up close direct interaction.

Space and layout management

In any display, portions of an infinitely large virtual canvas can be invisible to the user. On desktop-scale displays, various window- and desktop-management schemes let users effectively manage the available display area's space and layout. Space and layout management issues are different for large-scale displays: Although the displays can simultaneously display more data, some of it will be outside the user's focal visual field. The increased display space makes traditional space-multiplexing techniques such as overlapping windows less relevant. However, the increase introduces the need for appropriate techniques for managing, laying out, and fluidly accessing information in a nonspa-

 Published by the IEEE Computer Society

1 Up close interaction with a high-resolution multiprojector large-scale display.

tially overlapping manner over a broader display space spanning a larger view field.

Aided context switching

Desktop window-management systems provide various mechanisms for context switching between spatially overlapping applications and windows. On large displays, application windows typically overlap less and thus require different context-switching techniques. When interacting with large displays, users can switch context by simply walking toward the relevant part of the screen. Although this is a simple and easily understood interaction, it requires the user to physically move more than might be desirable. Moreover, it doesn't support sophisticated in-place context switching between multiple applications. Thus, we need mechanisms for switching between applications that appropriately leverage the unique affordances of the larger display space.

Canvas portal design elements

Canvas portals have four primary elements that users can interactively manipulate.

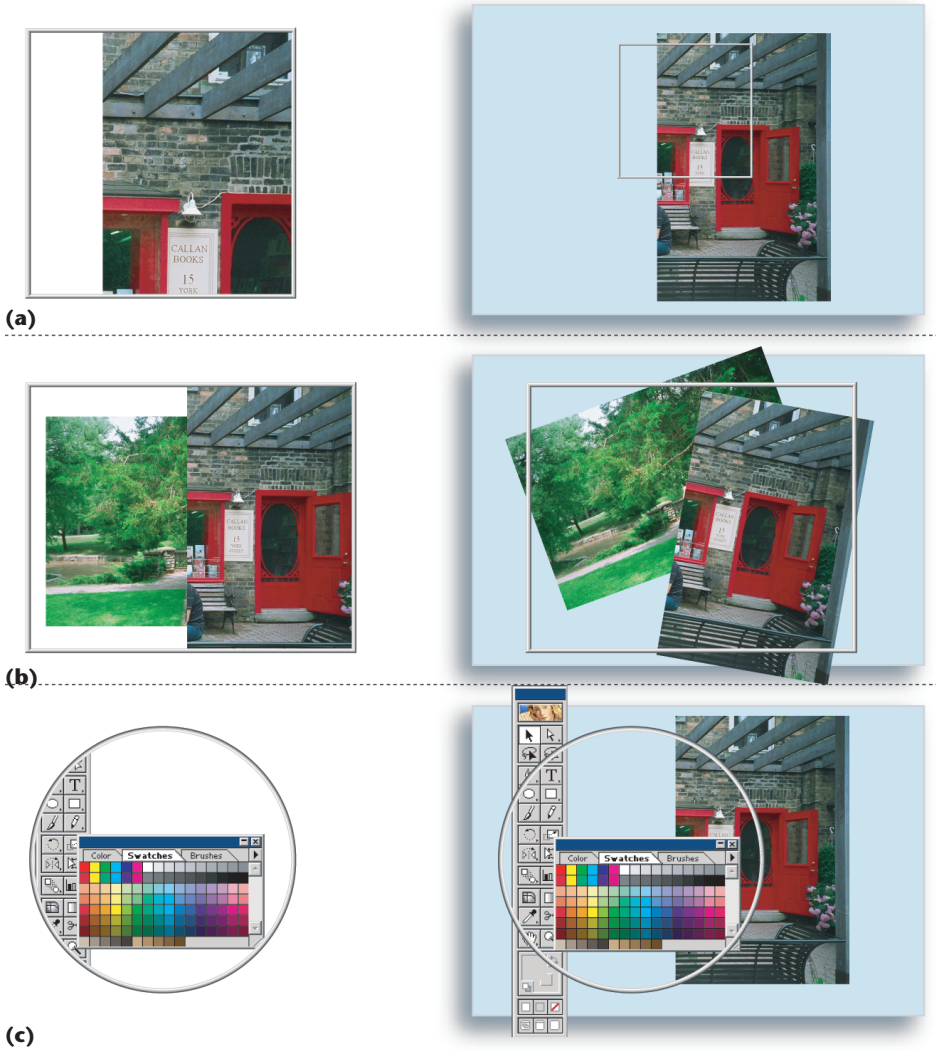
The *portal area* is the graphical representation of the canvas portal within which interaction occurs and can exist on the main display or on entirely separate devices such as PDAs or laptops. The portal area can have different shapes and sizes, and users can reposition it as desired. The portal area's shape is typically closely related to that of the focal area, such that the mappings between them are simple and easily understood.

The *focal area* is the remote part of the main display canvas on which the portal area's view is centered. Interaction with proxy elements inside the portal area is typically equivalent to interaction with the actual elements within the focal area on the main display. The system is designed to map input events occurring within the portal area to the focal area, either directly or with a transformation appropriate to the desired application.

Attributes describe the portal area's scale, semantics, state in time, and so on. In the simplest case, the portal area displays exactly the same content as the focal area. However, users might want to alter the portal area's display attributes such that it displays a transformed version of the focal area's content. By varying the portal area's attributes, the canvas portal acts like a magic lens[1] to the display's focal area. Manipulating the portal area's scale attribute, for example, makes the portal area act like an interactive zoom lens to the focal area. Similarly, manipulating different rendering attributes lets users view, for example, solid objects as wire frames within the portal area. Manipulating semantic attributes can limit the portal area's display to a certain type of object. For example, the portal area might display only user interface elements, thus enabling on-the-fly tool palette creation. Manipulating time can make the portal area display the focal area's state at a previous time instant. An attribute could even be the user to whom the portal belongs, letting different users maintain different views of the canvas layout. Figure 2 (next page) shows example canvas portals with varying shapes, sizes, and attributes.

Boundaries define the transition zone between the portal area and the main canvas. Unlike regular magic lenses, canvas portals support the passing of objects between the portal area, the main display canvas, and other canvas portals. If a user is moving an object on the main canvas and the center of movement (typically the cursor) crosses a portal area's boundary, the object transitions into the portal area and continues its movement inside the portal area's coordinate system. The inverse also holds. Transition boundaries can be the portal area's discrete physical limits, a transition area beyond the portal area, or widgets in the portal area that the user interacts with to initiate the transition.

Consider, for example, a canvas portal with a zoomed-in scale attribute in which the portal area's discrete border acts as the boundary, as in Figure 3. The cursor's center of movement is thus the center of translation and scaling of the objects being moved. When items cross a



2 Canvas portal shape and attribute example. The left column shows the portal areas of the canvas portals; the right shows the remote focal areas on the displays. (a) Rectangular canvas portal with scale attribute modified to act as a zoom lens to the focal area. (b) Rectangular canvas portal with modified time attribute displaying a previous state of the focal area's content. (c) Circular canvas portal with semantic attribute modified to show only user interface elements, creating a tool palette.

boundary, the system notifies the user in one of several ways. In one implementation, when parts of a moving item cross the portal boundary, the system casts a translucent rendering of these parts on the other side of the boundary, indicating that the user can extend the movement outside the current working reference frame, which can be the main display or the portal itself. Thus, users can access remote objects by directly interacting with them within the portal's reaches, or by moving the remote object nearer to them on the main canvas by crossing the boundaries.

By supporting the seamless transition of items between a canvas portal and the main display canvas, the framework lets us quickly rearrange items in and out of the portals without always having to refocus the portal as in a magic lens, thus providing an efficient layout management mechanism. It also enables interesting use scenarios, particularly when multiple canvas portals are active simultaneously. For example, after interacting in detail with an object in a canvas portal using a zoomed-in scale attribute, a user can push the object into a second zoomed-out canvas portal acting as a temporary space for working items without cluttering the main display area. This aided context switching is especially crucial to large-display interaction because alternating between context levels by simply viewing the display from different distances can prove cumbersome.

The four design elements let us instantiate canvas portals into a variety of existing and new interaction techniques. Each technique modifies design elements differently, including giving users control over some parameters while constraining others to a predefined value. For example, users can easily instantiate a Manhattan lens by fixing the portal area's position while letting the user reposition the focal area interactively.

Instantiating existing techniques in canvas portals

To illustrate how the framework could identify and address some limitations of current techniques, we instantiate work on alternative views for large displays in the canvas portal framework.

Dollhouse

Swaminathan and Sato[2] identify problems that arise because of the scale of large displays, including pointer movement and control challenges over large distances. They propose using a dollhouse metaphor: a small-scale model of the display and its contents to specify pointer movement in the large display. We can implement their technique as a canvas portal with a modified scale attribute whose focal area encompasses the entire screen. Movements inside the portal area cover larger distances than on the display, accelerating item selection. Although well suited for distance reaching, the technique lacks the potential for complex layout management and sophisticated context switching, because the dollhouse view uses a fixed scale factor and doesn't permit passing objects between views.

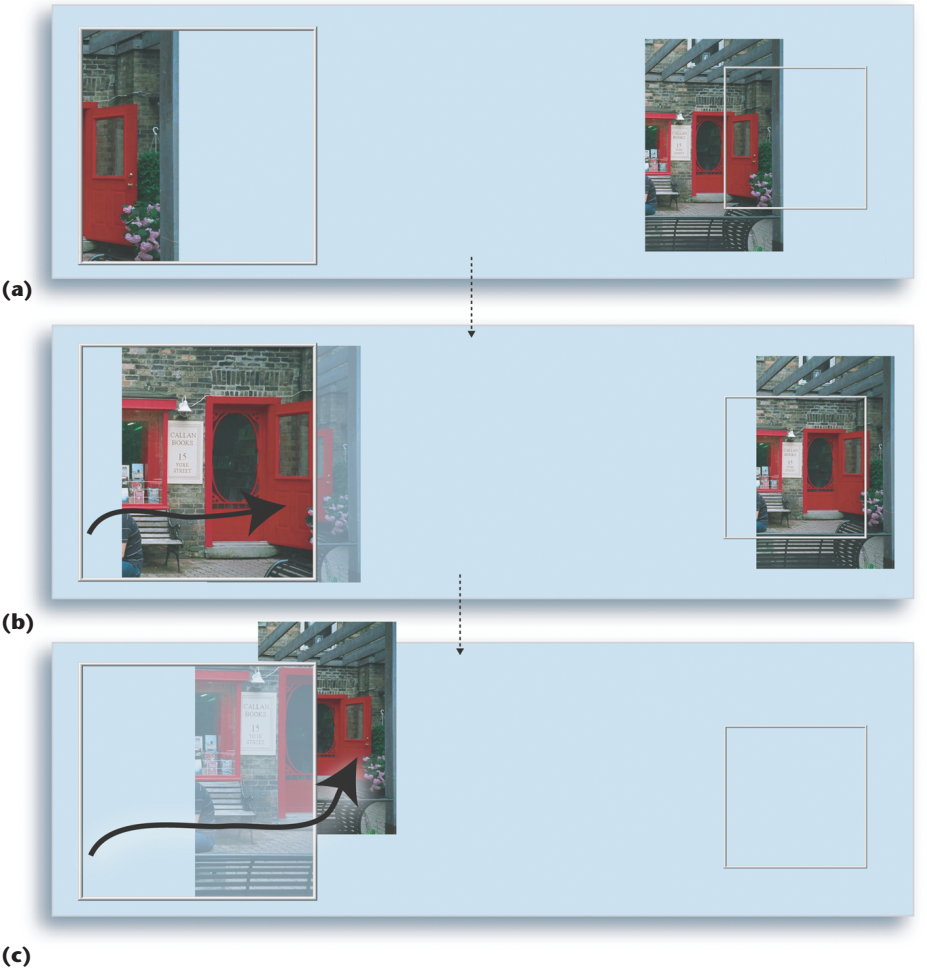
ZoomScapes

ZoomScapes[3] are regions of the screen with different zoom levels. Objects crossing ZoomScapes are scaled around the center of the cursor's movement in a con-

tinuous manner. We can implement ZoomScapes as canvas portals with portal and focal areas of fixed positions, scale attributes that are modified to reflect different zoom levels, and boundaries that support scale attribute transitions as objects cross between the canvas portals corresponding to the ZoomScapes. ZoomScapes is an excellent space-management technique. In its present form, however, we can't easily extend it to address distance reaching issues because it can't connect to a particular remote area of the virtual canvas.

Scalable fabric

More recently, several papers have described interesting techniques that let users define alternative work area views. Scalable fabric[4] enhances existing window managers using an area around the user's main focus to scale down windows and create user-defined, task-related groups. Selecting a task group in this area brings it into focus, replacing the previous group in the main display area. Although designed for regular desktop computers, scalable fabric areas are like noninteractive canvas portals with varying scale attributes and interchangeable focal and portal areas. In the context of high-resolution displays, scalable fabric could be enhanced by allowing users to interact with the scaled-down versions similar to the actual windows because the high resolution ensures that smaller-scaled versions of objects remain appropriate for detailed interaction.



3 Canvas portal boundary transitions. The images on the left are portal areas; those on the right are remote focal areas. Black arrows indicate user movement. (a) A canvas portal with a zoomed-in scale attribute. (b) The user moves an object inside the portal area. A semitransparent rendering of the object beyond the portal area indicates that the movement can be continued outside the portal area's boundary. (c) As the object passes the portal area's boundary, it transitions to the main display and is no longer available in the focal area. A semitransparent rendering of the object inside the portal area indicates that the user can transition the object back inside as long as the user maintains the drag event.

WinCuts

WinCuts[5] augment window managers by letting users acquire and interact with alternative views of arbitrary regions of existing windows. Unlike WinCuts, canvas portals aren't attached to specific windows or parts of them. Thus, when a window is hidden behind other windows on the canvas, it's also hidden within the canvas portal. Nevertheless, canvas portals with an interactively modifiable time attribute can provide functionality similar to WinCuts because they can focus on display layouts in which the desired window is in focus. Alternatively, imagine a canvas portal with a user-specified top window attribute that ensures a particular window or portion thereof is always in focus within the portal area. WinCuts are effective as context remote area view mechanisms, but they don't accommodate layout management.

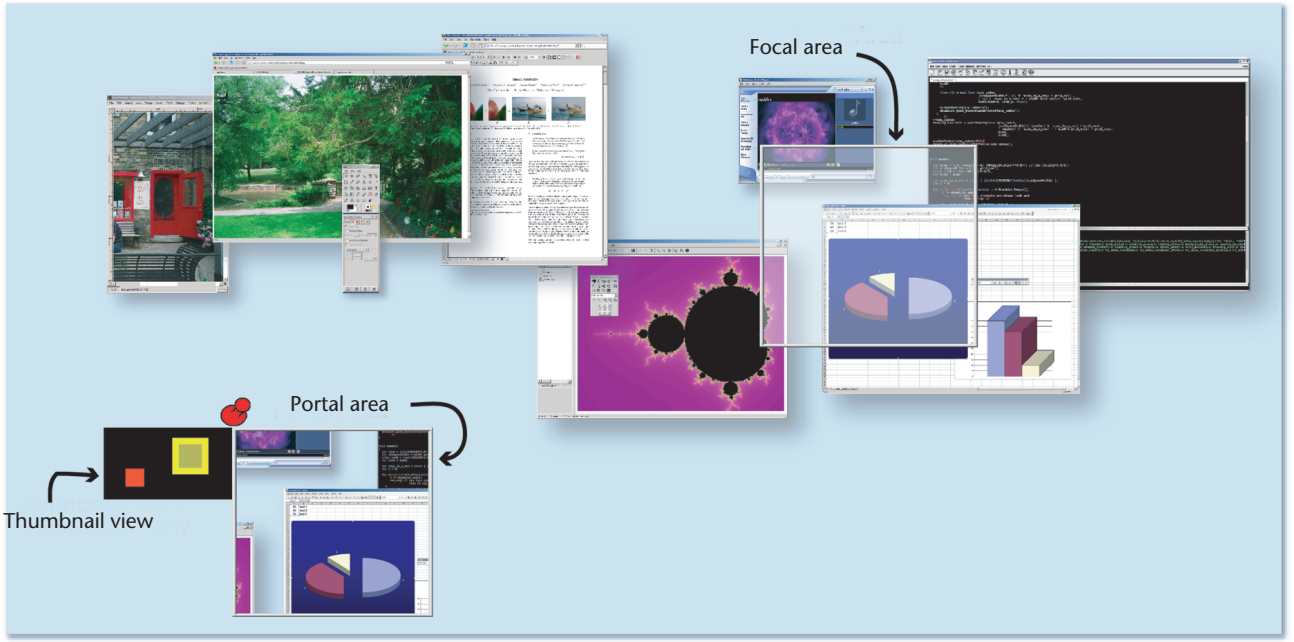
Frisbee

The Frisbee[6] is a widget that acts as a telescope to a remote area on the display. Users manipulate remote items by interacting with their proxies within the Frisbee's main area and reposition items on the main display by moving them through specified transfer channels. Users can manipulate the remote focal area's position and the scale of the items within the Frisbee widget. We can instantiate the Frisbee within the canvas portals design using a single canvas portal with a modifiable scale attribute and define boundaries in specific locations around the portal area.

Table 1 (next page) summarizes how we represent these techniques with their current functionality within the canvas portal design space. Note that we've focused on the conceptual representation of canvas portals and its various instantiations rather than the fine details of each instantiation. Such details typically involve issues

Table 1. Instantiation of existing techniques.

Technique	Attribute	Portal area	Focal area	Boundary
Dollhouse	Scale	Scaled display shape	Entire display	No crossing
ZoomScapes	Scale	Area of display width	Areas of scaled display width	Smooth crossing when traversing boundary
Scalable fabric	Scale	Rectangles along the display's border	Areas along the display's border	Smooth crossing when entering the focal area border
WinCuts	Time	Custom rectangle	Area encompassing desired part of a window at a particular time	No crossing
Frisbee	Scale	Round shape	Custom round area	Smooth crossing at transfer channels



4 The ScaleView portal is at the bottom left corner of the display. The red pin icon on the top-left corner indicates that the ScaleView portal can be moved. In this instance the portal is not moving, thus the pin icon is "pinned." The black rectangle at the side represents a thumbnail view of the virtual display canvas, within which the yellow rectangle indicates the focal area's position on the display canvas, and the red rectangle indicates the ScaleView portal's position. An outline rectangle on the main display canvas represents the focal area.

that are implementation dependent but not fundamental to the underlying idea for each technique, such as interface elements for modifying parameters.

New canvas portal instances

We developed several new techniques within the canvas portal design space and have implemented and tested them in our laboratory. Preliminary user feedback has led to ongoing refinements.

We prototyped the implementations on a back-projected 16-foot wide by 6-foot high screen. Eighteen projectors (1,024 × 768 resolution each) generate the imagery in a 6 × 3 tiling, as Figure 1 shows. A cluster of 18 workstations drives the projectors. Software is written in C++ with Chromium (http://chromium.sourceforge.net) providing graphics rendering over the cluster. Our techniques are designed for pen and touch-sensitive displays, with finger tapping or pen down providing a single button event.

Our implementation of the following techniques uses an object-oriented approach. We first created a generic canvas portal class incorporating the design elements. Each technique is an instance of the generic canvas portal class, inheriting the functional characteristics of the parent class and extending them with additional routines to enable specific interactions.

ScaleView portals

A ScaleView portal provides views of the display's areas at different scales. The ScaleView portal, shown in Figure 4, appears as a window-like widget onscreen and has three user-controllable parameters: the focal area's position on the virtual canvas, the value of the portal area's scale attribute relative to the focal area's, and the portal area's position on the virtual canvas.

Users can reposition a ScaleView portal on the virtual canvas by clicking and dragging a pin icon (see Figure 4). A FastSlider[7] invoked from a marking menu attached

5 Grouping of multiple ScaleView portals: (a) stacked, (b) side-by-side, and (c) cube layout. Clicking on a portal brings it to the top.

to the portal adjusts the scale factor. Users can alter the focal area's position in two ways. The most direct and precise method is to select the "change focal area position" item from the portal's marking menu and then click the desired location on the virtual canvas. This method, however, can be inconvenient if the desired focal area is on a difficult-to-reach part of the screen, or impossible to achieve if the area of interest is on a part of the virtual canvas that isn't currently displayed on screen.

In the second technique, the user selects the focal area at a coarser granularity using a thumbnail representation of the entire virtual canvas. This thumbnail is attached to the top left corner of the canvas portal and shows an iconic representation of the focal area as well as the portal's position on the larger canvas. The user drags the icon representing the focal area to reposition it. As the user drags the icon, the system provides context by highlighting the corresponding region on the main canvas with a semitransparent overlay. This approximate way of changing the focal area lets the user operate the ScaleView portal without having to move around the screen. The user can also reach areas at the screen's extremities or areas of the virtual canvas not visible on screen.

Our implementation lets users pan the entire virtual canvas across the display screen. To retain a canvas portal view when panning the virtual canvas, users can "pin" a ScaleView portal to a particular area of the virtual canvas, as opposed to an area of the physical display screen. For example, if the user focuses and pins a ScaleView portal on a group of windows, that focal area stays unchanged as the user pans the virtual canvas.

The ScaleView portal's boundaries are its physical borders. If the user drags an item out of the portal, the system removes the item from the remote focal area and positions it on the main virtual canvas at the position of the user's actions. Conversely, a user can drag an item from the main canvas into the ScaleView portal, which automatically moves it to the portal's remote focal area.

To transfer a ScaleView portal's view to the entire display screen, a selection from the portal's marking menu warps the entire screen's view to match that of the canvas portal, centered at the portal's location. This way, the portal lets users specify viewpoint transformations for the entire screen by matching the focus and portal areas.

ScaleView portals let users interactively adjust the scale factor between portal area and focal area, so users can easily select remote or small-sized targets using zoomed-out or zoomed-in ScaleView portals. This enables ScaleView portals to act as remote reaching mechanisms.

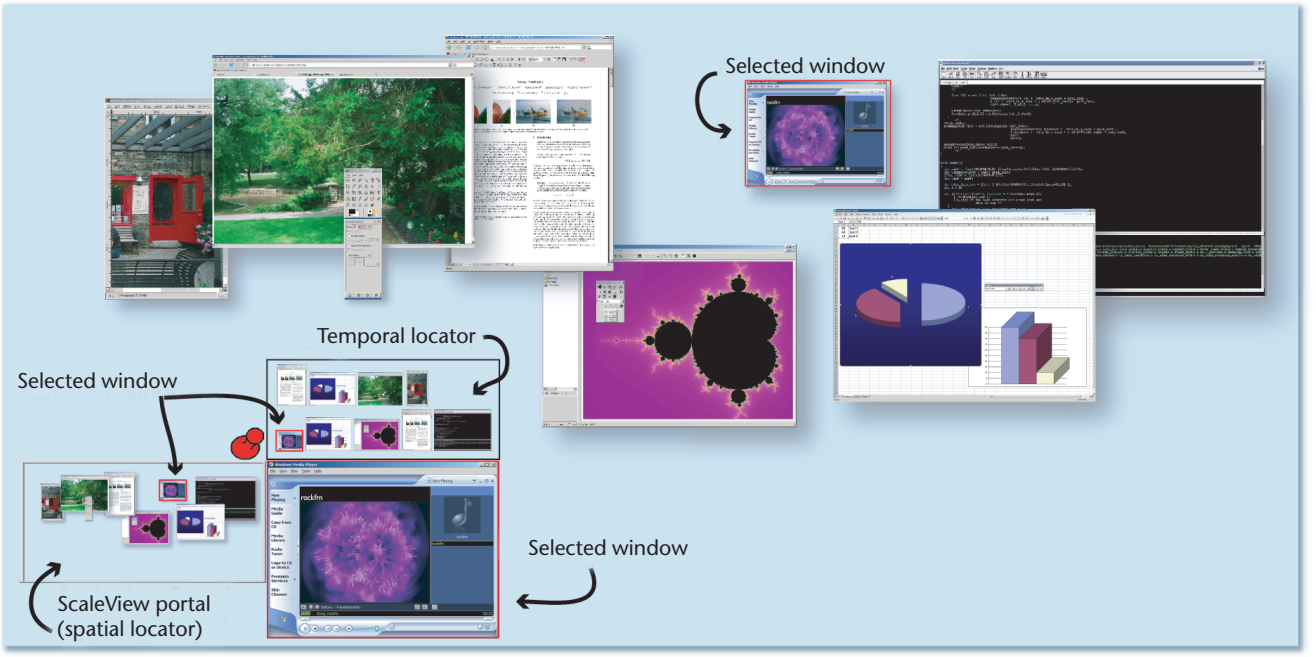
The screen in our system is assumed to be touch sensitive, with all interaction occurring close to the screen. However, when interacting with large-scale, high-resolution displays, users might want to interact from slightly further away, perhaps using different input modalities. Although moving away from the screen gives users a better overview of the screen without specially designed widgets, the functionality provided by ScaleView portals can still be useful. For example, they provide additional views so users can switch context less frequently or faster when moving between tasks or applications. They also provide detail and overall views of the virtual canvas simultaneously, which is useful at any scale of interaction. Finally, users can group items and move them seamlessly across different scales and views even when interacting at a distance. ScaleView portals thus enable both context switching and layout management.

In our implementation, ScaleView portals are displayed on the same canvas as the large display (although they can also reside on a remote device). Large numbers of ScaleView portals are cumbersome to relocate, require considerable screen space, and can lead to overlapping. To alleviate these issues, we allow users to group multiple ScaleView portals into stacked, side-by-side, or cube layouts, such as those in Figure 5. Users switch between portals in a stacked or cubed group by either touching a constituent portal, which brings that portal to the top with an animated transition, or by expanding the stack and cubed view into a side-by-side layout, where the user selects a new portal by clicking on it. Although we've only implemented three grouping techniques for ScaleView portals, many other groupings are possible.

Having multiple ScaleView portals active can make it difficult to distinguish which portal area corresponds to which focal area on the virtual canvas. To mitigate this, each time a transition between multiple portals occurs, we currently display a connecting line between the portal and focal areas. Nevertheless, as the number of portals increases, several of them might point at similar areas of the virtual canvas, making it difficult to distinguish them. In future versions, we'll explore labeling the portals and providing a tabbed layout of labels similar to that available in some desktop window managers.

Window portals

Window portals, illustrated in Figure 6 (next page), are a variant of canvas portals providing quick access to, and switching between, application windows on a large-



6 Window portal on a large display. An overview region on the left of the Window portal—a ScaleView portal with the entire screen as the focal area—serves as a spatial locator. A temporal locator (timeline) display is on top. The selected object is on the bottom right of the widget. Red borders around the selected object in all views help maintain context.

scale display. The main overview region of the Window portal is functionally equivalent to a ScaleView portal with a focal area covering the entire virtual canvas (that is, acting as a zoomed-out view of the entire canvas). When a user clicks on an item in the overview region, the system expands the item, displays it as an active object next to the overview region, and gives it a red border to indicate its special status. The user can interact in detail with the selected item or select a new one. As the user selects objects over time, a thumbnail representation of all previously selected items is displayed in a timeline over the current item. Clicking on a thumbnail turns the associated item into the active item. Thus, whereas the Window portal's overview region acts as a spatial locator of items on the main canvas, the timeline region acts as a temporal locator for recently used items.

In addition to serving as a context-switching mechanism between an overview and detailed interaction with a specific object, Window portals provide spatial and temporal shortcuts to potentially remote objects. Unlike the overview areas of virtual desktop managers, the Window portal's overview is fully interactive, allowing coarse actions in the overview and fine-grained actions in the detailed application window.

On the standard Microsoft Windows desktop, the Alt-Tab key combination lets users switch between application windows. When interacting with a large display without a conveniently accessible keyboard, Window portals provide similar spatial and temporal switching between active windows. This functionality can also be useful in smaller displays lacking an easily accessible keyboard, such as tablet PCs in a slate configuration.

If we enable semantic filtering of items in the Window portal's overview region, we could restrict the display to, for example, user interface widgets. Thus, we could create on-the-fly palettes of interface widgets that we could move around the screen to operate multiple applications from a single locale. Although semantic filtering isn't currently implemented, users can create palette portals themselves.

Given that the Window portal consists of a zoomed-out view of the screen, issues related to selecting and distinguishing between small targets arise. Although semantic filtering can limit the selection space, the potential number of items (overlapping or not) can still be large. Moreover, semantic filtering doesn't address the issue of selecting small targets. Allowing dynamic zooming or expanding of targets is a potential solution.

Overlapping windows on the canvas can be problematic in Window portals as in any other multiwindow management system. Current solutions include tabs, peeling, and multiblending. We've implemented an alternate approach that fans out a group of overlapping windows when a user clicks on any member of the group, in a manner similar to the widgets presented elsewhere.[8,9] Users can select items from this fanned-out display, or collapse the group again. By default, we treat overlapping objects as a group. This enables not only the fan-out operation, but also lets users move the entire group as a whole within the Window portal. We've implemented this grouping feature across all our widgets.

Division bands

ScaleView portals and Window portals provide alternative ways to view and access data while essentially preserving the main virtual canvas's overall view and layout. Division bands, shown in Figure 7, also provide alternate views of the virtual canvas, but unlike the

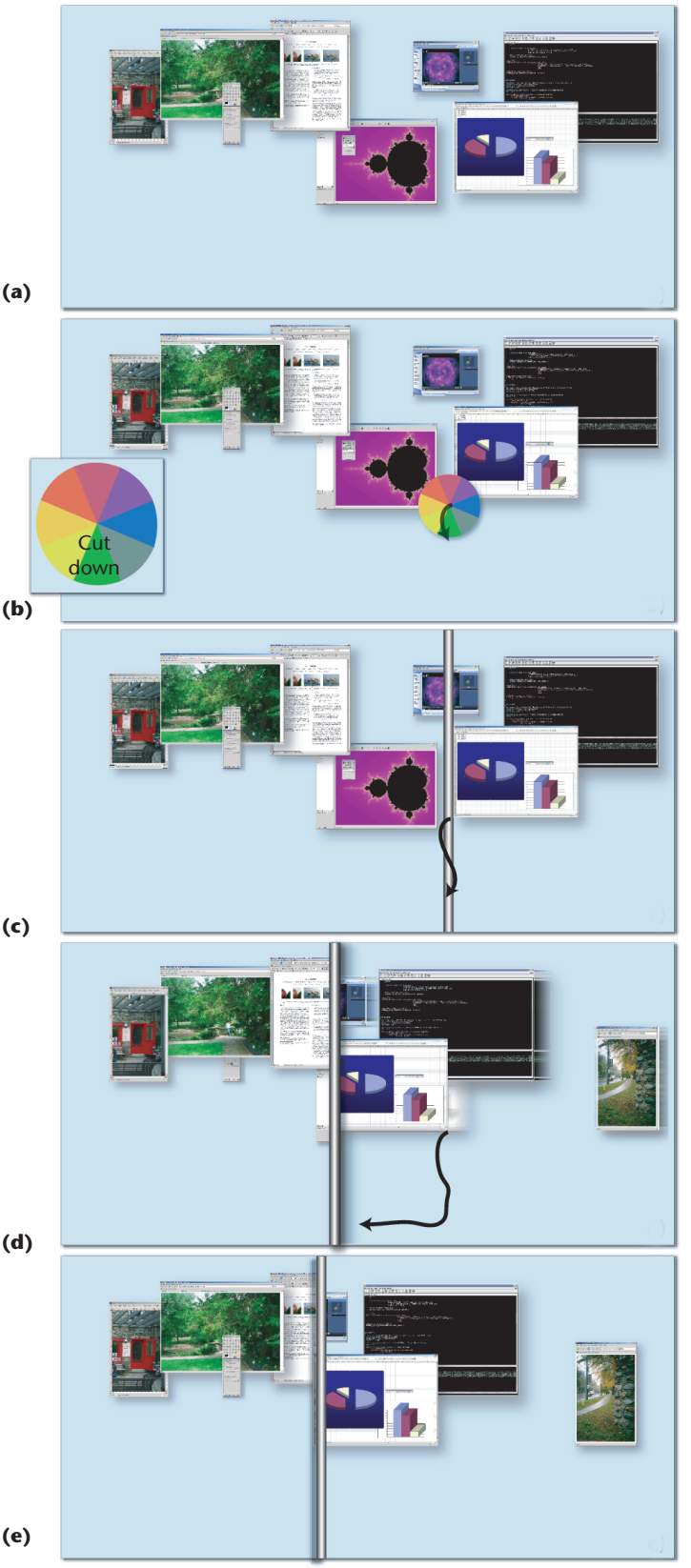
other techniques, they temporarily disrupt the virtual canvas' overall view. They act much like a cutting tool that virtually slices the canvas along specified vertical and horizontal directions and lets users drag the cut portions around to quickly reveal more or less of certain parts of the virtual canvas. In essence, they're ScaleView portals with a fixed focal area and a fixed portal size equal to the screen, but with boundaries that can be quickly repositioned.

Using a marking menu, the user specifies the cut position (the menu's invocation point) and direction (the mark used in the menu selection). We deliberately placed the cut commands in menu locations that would require selection marks in the corresponding directions to facilitate a fluid, combined specification of command and parameter. After the user specifies the cut position and direction, a vertical or horizontal (depending on the specified direction) division band widget appears on screen, and the system attaches one of the cut pieces of the canvas to it, based on the specific mark on the menu. If the user specifies "cut-down" and moves to the left, for example, a vertical division band appears with the right side of the canvas attached to it. This division band is now attached to the user's pointer and can be moved. By dragging the division band left to right (or right to left) on screen, the user can shrink (or expand) the attached portion of the canvas. A quick flick-and-release motion of the cursor dismisses the division band. If the user releases the division band without the flicking motion, the division band remains on screen for subsequent reselection. Thus, division bands let users quickly drag a part of the screen toward them for viewing and/or manipulation. The ability to quickly dismiss the division band with a flicking gesture allows for transient, quick views of remote portions of the screen, much like pulling on a spring-loaded window blind.

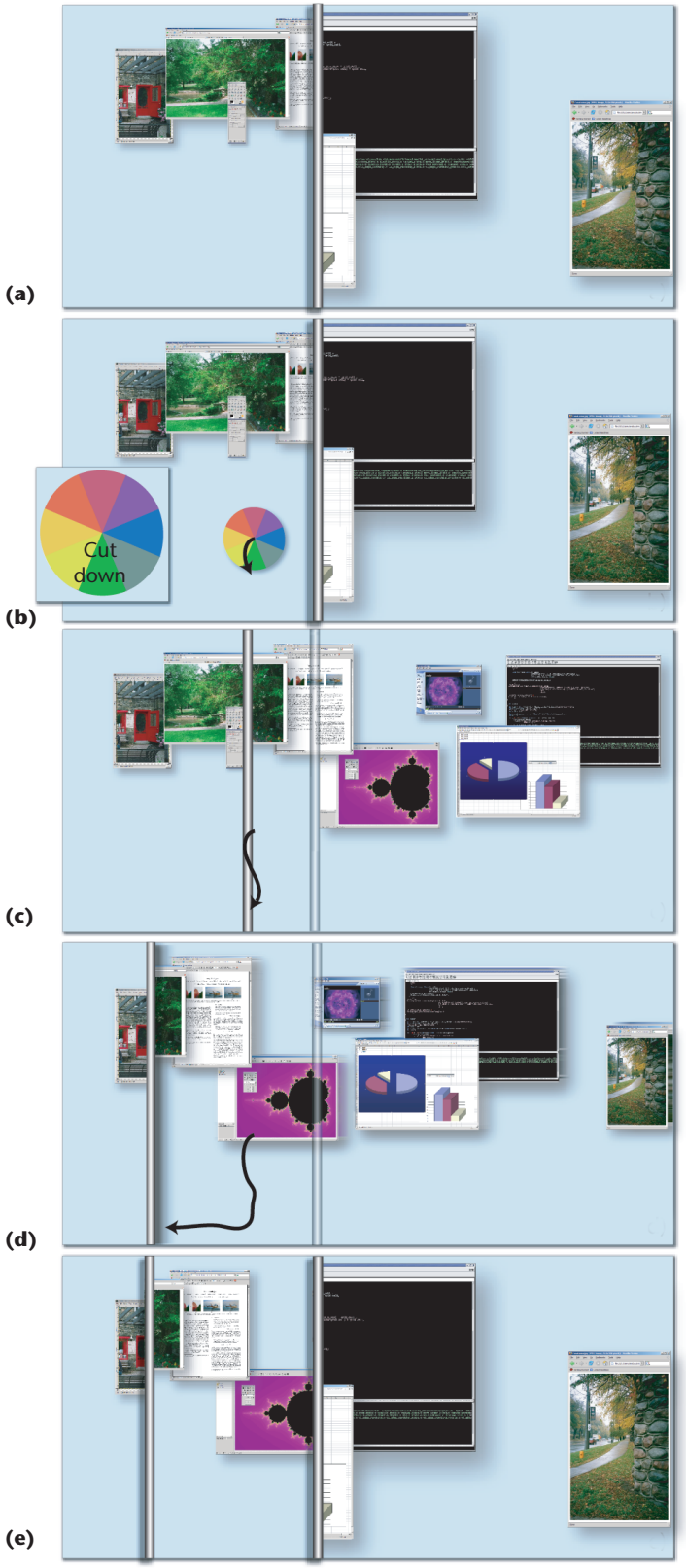
Users can position multiple division bands of different orientations and directions on the screen, as Figure 8 illustrates. The system presents multiple, pinned division bands of different orientations and directions in a fixed ordering. It places horizontal bands on top of vertical bands because we've found they're more likely to be followed by a permanent pinning action. Horizontal bands seem most useful for creating new space onscreen by dragging unused portions of the virtual canvas into view from the top or the bottom. In contrast, vertical bands tend to be used more transiently to perform quick glancing actions.

The quick creation and dismissal feature of division bands makes them well suited for fast, glancing actions at remote content or parts of the virtual canvas currently off screen. With the pinning option, division bands can also serve as persistent shortcuts to remote areas of the screen or as virtual desktops, similarly to the Flatland flipcharts.[10] As with all canvas portals instantiations, our implementation lets users move objects between the main canvas and portions of the canvas attached to division bands. Division bands thus can act as both layout management and remote reaching mechanisms.

Newly created division bands have the same scale factor as the regular canvas, but we provide a menu option to enable users to change the scale, allowing for the cre-



7 Single division band creation. (a) Initial state of the board. (b) The user invokes the marking menu. (c) The system creates a new vertical band at the location of the marking menu and attaches it to the user's cursor. (d) Pulling the band to the left reveals more of the canvas to the right. (e) Final state of the board after the user releases the band. If the user makes a quick flicking action instead, the band is discarded. (Black scribbles indicate user strokes.)



8 Second division band creation. (a) One division band is already present. (b) The user invokes the marking menu and (c) creates a second vertical band at the menu location that's attached to the user's cursor. As long as the user is moving the new band, it overlaps the existing band. (d) The user moves the band to a new location and (e) releases it. The bands are sorted so that they're both visible (here the original band is positioned on top of the newly created one).

ation of a form of ZoomScapes.[3] Because bands can have their own zoom factors, users can move items back and forth between bands for different visualization effects. For example, users can store objects in zoomed-out bands to save space or bring them into zoomed-in bands for detailed inspection. Although we haven't implemented ZoomScapes' elegant transition of groups of objects, we could incorporate this characteristic of the canvas portal framework's boundary into the different techniques.

Division bands are similar to the split-window mechanism of recent word processors in terms of splitting of the canvas, but they also provide transient and fluid ways to specify, invoke, and dismiss the splits, thus improving usability. Division bands also enable sophisticated context switches between bands.

A potential problem is the visual effect of virtually cutting of the canvas. This isn't an issue for single-user interaction, because the cutting originates from the user's own actions and is unlikely to cause confusion; however, if multiple users interact with the display simultaneously, division bands could prove confusing, and solutions need to be developed.

We've observed that horizontal division bands are most often used for facilitating the management of working and storage spaces, whereas vertical bands are most often used for quick glancing actions or fast shortcuts and their lifespan on the canvas is more limited. Although we've sorted the display of multiple division bands based on this observation, we intend to investigate patterns of use of the different direction bands in more detail and create a sorting and visualization mechanism better tailored to user behavior.

The direct creation mechanism of division bands and the bands' ephemeral nature indicate that they can be used to take state snapshots of the screen and act as a form of history of the virtual canvas, thus leveraging canvas portals' time attribute. However, we'll need to solve the visualization challenges of depicting and navigating groups of such history bands related to their spatial and temporal nature.

Future work

Our work has focused on a usage scenario in which the user interacts up close to a touch-sensitive screen capable of only 2-DOF x-y input, with a one-button event. However, we could significantly extend many of our techniques by using more sophisticated input devices or finger and hand gestures. Even a simple input enhancement, such as detecting finger hover over the screen's surface, could expand the interaction vocabulary. For example, we could use hover to provide transient magnification of small-scale items in ScaleView portals. It could also help simplify the interactions that currently require continuous press-and-hold dragging actions, such as in division bands.

Although the focal areas of the described techniques have so far been rectangles, we'll let users create other shapes as well. We expect that more free-form shapes could help users remember item groupings. Similarly, skinny rectangles could be more useful for capturing tool or menu bars. Division bands could also potentially benefit from cuts beyond straight lines.

Because we've focused on single-user techniques, our methods share the same instances of onscreen objects. We use wall-sized displays, however, so it's desirable that the techniques be extendable for multiuser interaction. Although multiple users could use some of our tools simultaneously without interfering much with one another (such as the ScaleView and Window portals), adding functionality for context-sensitive objects would require enhancing others, such as division bands, to support multiple users. This area clearly requires significant future research. ■

Acknowledgment

We thank the members of the Dynamic Graphics Project lab (http://www.dgp.toronto.edu) at the University of Toronto.

References

1. E. Bier et al., "Toolglass and Magic Lenses: The See-Through Interface," *Proc. ACM Siggraph Conf.* , ACM Press, 1993, pp. 73-80.
2. K. Swaminathan and S. Sato, "Interaction Design for Large Displays," *Interactions*, vol. 4, no. 1, 1997, pp. 15-24.
3. F. Guimbretière, M. Stone, and T. Winograd, "Fluid Interaction with High-Resolution Wall-Size Displays," *Proc. ACM Symp. User Interface Software and Technology* (UIST), ACM Press, 2001, pp. 21-30.
4. G. Robertson, "Scalable Fabric: A Flexible Representation for Task Management," *Proc. Advanced Visual Interfaces* (AVI), ACM Press, 2004, pp. 85-89.
5. D. Tan, B. Meyers, and M. Czerwinski, "WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space," *Proc. ACM CHI Conf. Human Factors in Computing Systems*, ACM Press, 2004, pp. 1525-1528.
6. A. Khan et al., "A Remote Control Interface for Large Displays," *Proc. ACM Symp. User Interface Software and Technology* (UIST), ACM Press, 2004, pp. 127-136.
7. M. McGuffin, N. Burtnyk, and G. Kurtenbach, "Fast Sliders: Integrating Marking Menus and the Adjustment of Continuous Values," *Proc. Graphics Interface*, A K Peters, 2002, pp. 35-42.
8. M. McGuffin, L. Tancau, and R. Balakrishnan, "Using Deformations for Browsing Volumetric Data," *Proc. IEEE Visualization*, IEEE CS Press, 2003, pp. 401-408.
9. F. Vernier, N. Lesh, and C. Shen, "Visualization Techniques for Circular Tabletop Interfaces," *Proc. AVI Advanced Visual Interfaces*, ACM Press, 2004, pp. 257-263.
10. E. Mynatt et al., "Flatland: New Dimensions in Office Whiteboards," *Proc. ACM CHI Conf. Human Factors in Computing Systems*, ACM Press, 1999, pp. 346-353.

Anastasia Bezerianos *is a PhD candidate in computer science at the Dynamic Graphics Project at the University of Toronto. Her research interests include interaction techniques for large-scale displays for single and multiple users. Bezerianos has an MS in computer science from the University of Toronto. Contact her at anab@dgp.toronto.edu.*

Ravin Balakrishnan *is an assistant professor in the Department of Computer Science, University of Toronto, where he codirects the Dynamic Graphics Project laboratory. His research interests include human–computer interaction and interactive computer graphics. Balakrishnan has a PhD in computer science from the University of Toronto. Contact him at ravin@dgp.toronto.edu.*