

CoR²Ds: Context-Rooted Rotatable Draggables for Tabletop Interaction

Chia Shen¹

¹Mitsubishi Electric Research Labs
201 Broadway
Cambridge, MA, 02139, USA
{shen, forlines}@merl.com

Mark S. Hancock^{1,2}

²Department of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, AB, Canada T2N 1N4
msh@cs.ucalgary.ca

Clifton Forlines¹Frédéric D. Vernier³

³University of Paris 11,
LIMSI-CNRS, BP 133, 91403
Orsay, France
Frederic.Vernier@limsi.fr

Abstract

We present a new popup widget, called CoR²Ds (Context-Rooted Rotatable Draggables), designed for multi-user direct-touch tabletop environments. CoR²Ds are interactive callout popup objects that are visually connected (rooted) at the originating displayed object by a semi-transparent colored swath. CoR²Ds can be used to bring out menus, display drilled-down or off-screen ancillary data such as metadata and attributes, as well as instantiate tools. CoR²Ds can be freely moved, rotated, and re-oriented on a tabletop display surface by fingers, hands, pointing devices (mice) or marking devices (such as a stylus or light pen). CoR²Ds address five issues for interaction techniques on interactive tabletop display surfaces: occlusion, reach, context on a cluttered display, readability, and concurrent/coordinated multi-user interaction. In this paper, we present the design, interaction and implementation of CoR²Ds. We also discuss a set of current usage scenarios.

Categories & Subject Descriptors: H.5.2. [Information interfaces and presentation]: User Interfaces – Interaction Styles, Graphical User Interfaces (GUI).

General Terms: Design, Human Factors, Performance.

Keywords: Drag and drop, popup menu, tabletop display, Computer Supported Cooperative Work (CSCW).

TABLETOP DISPLAY INTERACTION

Multi-user tabletop environments present new challenges in the design of interaction techniques. In particular, five problems are prominent in user interfaces for large direct touch multi-user tabletops: occlusion, reach, contextual-association, readability, and concurrent interaction.

Occlusion – User interactions on direct touch surfaces can cause occlusion by the input device. This occlusion is compounded when a popup (menu or a tool bar) is brought out on top of a document, since the document is occluded by both the input device and by the popup.

Reach – Since a large display surface can be wider than a person's arm length, situations may arise where one user needs to pass data or tools related to a document within

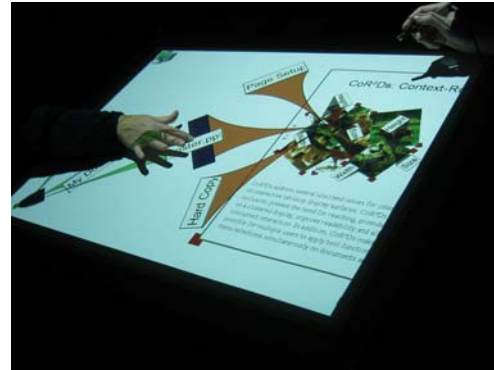


Figure 1. Two users employ CoR²Ds in a poster design scenario on an interactive tabletop.

reach of a user on another side of the display, but may not wish to pass the document itself. These tasks must be supported with movable tools and menus while not losing the original positioning of a document in the display space.

Contextual Association – As with non-interactive tables, tabletop displays have a tendency to become cluttered. Multiple users each contributing to the information that is displayed can exacerbate the problem. It can be difficult to determine which objects are being affected by a particular action of one of the users. An interaction technique must support a contextual association between the action or operation and the effected objects or operands.

Readability – Users may interact with the user interface from different angles and sides of the interactive table, thus conventional rectilinear text display is not necessarily readable by all users. An interaction technique must support some mechanism of rotation to improve readability.

Concurrent Interaction – On large direct touch surfaces, more than one user can be interacting with many applications, documents or objects simultaneously. Existing interaction techniques involving more than one discrete action (e.g. menus, combo boxes, modal dialogues) do not consider concurrent interaction and possible interference [6] between users. One user's actions should not cause the system to alter the display in such a way that it interferes with another user's work. Interaction techniques for multi-user tabletop displays should minimize unpredictable "behind-the-scenes" effects and allow the users to control

their coordinated interaction. The technique should also allow cooperative completion of any interaction.

We present a new popup interaction widget called CoR²Ds that has been designed to address these five issues.

RELATED WORK

Popups are fundamental and frequently used in modern computer interfaces. Menus and tools are two of the most commonly implemented popups, such as pull-down menus.

To increase efficiency in using on-screen tools, Bier et al. [2] proposed a see-through user interface widget called Toolglass. In order to allow free positioning of a tool while enabling efficient one-handed operation, Fitzmaurice et al. [4] implemented tracking menus. When the user's pointing device reached the edge of the tool container, the entire tool container started to follow the motion of the pointing device. Once the pointing device left the edge and was inside the tool container, the user could select a tool element for operation.

This previous work was intended for use on a desktop, a laptop or a tablet PC user interface. Given the typical size of these displays and their common use as single-user devices, only one Toolglass or tracking menu is normally active on the display surface. Moreover, each of these tools appears on or very near the object that it will affect, thus feedback for contextual association is not required.

Recent advances in large form factor interactive displays of wall and table sizes have brought forth research on new interaction techniques [1, 5]. The drag-and-drop technique is commonly used in desktop interfaces. With this method, the user drags an on-screen object onto another on-screen object. Baudisch et al. [1] extend this idea for large displays with drag-and-pick and drag-and-pop so that when dragging begins, the target objects automatically (thus reactively) extend toward the source object to minimize distance and prevent the user from crossing screen boundaries. This interaction technique solves the issue of reach for direct input devices by bringing icons to the user, but the source object must still be moved and the interaction only allows for one action to be performed for any source/target pair with a single movement.

There are two key differences between this technique and CoR²Ds: (1) CoR²Ds specifically address the five problems discussed above that arise in multi-user, co-located, collaborative interaction on horizontal displays; and (2) CoR²Ds afford proactive interactions, whereas techniques such as drag-and-pop/pick support a light-weight reactive interaction style. In proactive interaction, all the on-screen effects are under user-control, but in reactive interaction, some display changes are system instantiated (such as the automatic popup proxy extension of target objects towards the source object). This is an important distinction in the design for multi-user applications – with more than one person interacting on a co-located display, user-controlled on-screen interactions will afford better coordination and

collaboration, minimizing disturbance of co-workers' work spaces, even though it may require "heavier-weight", user-controlled interactions (rather than transient light-weight, system-controlled interactions).

Another interaction technique for multiple large screen displays is the Frisbee technique [5]. This implementation provides a solution for both reach and concurrent interaction by allowing a user to interact with a distant object through a "telescopic" view, leaving a second user's local workspace unaltered. However, this technique does not deal with issues of readability and clutter, nor does it provide contextual information about the location of the operand.

We have arrived at the conceptual design of CoR²Ds from observations on the usage of several multi-user tabletop systems. In particular, several applications constructed with DiamondSpin [11] explored popup menus on tabletops [7, 8, 9, 10]. As more multi-user collaborative tabletop applications have started to appear, the need for a generalized model for collaborative interaction on tabletops, CoR²Ds, naturally emerged.

SCENARIO: PAGE LAYOUT

A customer, a typesetter and a graphic designer gather around a tabletop display to discuss the design of the customer's poster. The customer plugs in his memory stick containing the photographs and text for the poster. An icon for the memory stick appears on the table which the customer double-taps with his finger to bring up a set of CoR²Ds that represent each file on the memory stick. The customer then drags the poster file CoR²D to the CoR²D labeled "Current Poster" in the middle of the tabletop that has been brought up by the typesetter. The photographs and text from this poster file now appear in the poster document. The graphic designer remarks that the poster would benefit from a particular graphic that she accesses by traversing a set of CoR²Ds representing the file structure on the tabletop display. She asks the typesetter to pass her the "Current Poster" CoR²D. The typesetter drags the CoR²D toward the graphic designer who drags the CoR²D for the graphic file onto it. The new graphic shows up in the poster document (without having moved the object itself within reach). The three continue to discuss and modify the layout and font choices for the poster, moving and rotating the pictures and editing the text as necessary.

CONTEXT-ROOTED ROTATABLE DRAGGABLES

CoR²Ds are interactive callout popup objects that are visually connected (rooted) at the originating digital object by a semi-transparent swath. The originating graphical object can be a text document, an image, the entire display background, or even another CoR²D. CoR²Ds can be used to bring out menus, display drilled-down or off-screen ancillary data such as metadata and attributes for maps and images, as well as to instantiate tools (e.g., graphics tools, text formatting tools). CoR²Ds can be moved, dragged,

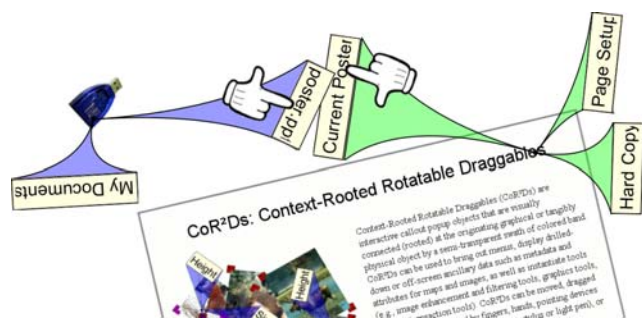


Figure 2. Concurrent Interaction on a large tabletop – Multiple independent CoR²Ds can be used in parallel by one user (with two hands) or multiple users to cooperatively complete a single interaction. CoR²Ds address the problems of occlusion, reach, contextual-association and readability.

rotated, resized, and re-oriented (translated and rotated in one motion). In addition, a single CoR²D or a pair of CoR²Ds can be used by multiple users to cooperatively complete a task. A CoR²D can be dragged to a different part of the display surface by one person at the table while operated on by another person, thus facilitating multi-user operations.

CoR²Ds are designed to allow the operator and the operand to be displaced at a visual distance, eliminating on-object occlusion, and enabling distant operation without losing the visual cue of the object that the menu or tools come from, which may be both at a distance on the display and partially hidden in a cluttered display.

We introduce a novel technique for the visual feedback of the context of a CoR²D. Our technique allows a CoR²D to be dragged away from the originating object without losing visual feedback of the context of its activation. In order to provide a multitude of operations, we also demonstrate two methods of placing multiple CoR²Ds associated with a single on-screen object. We have implemented CoR²Ds using the DiamondSpin toolkit [11], so that text items in CoR²Ds are readable from all sides of the display.

Visualization of Context

When a CoR²D is moved away from its starting location, visual feedback is provided to maintain the original context. This feedback is drawn between the current position of the CoR²D and its point of activation. The swath is a curved triangle with the first vertex of the triangle being at the

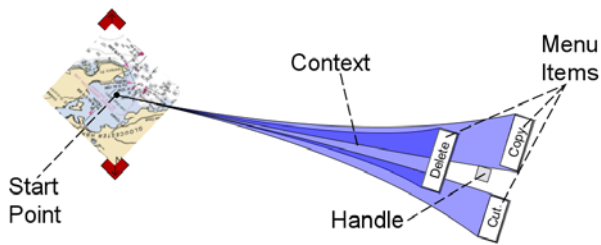


Figure 3. Grouped CoR²Ds move in unison by dragging the handle, while maintaining context with a transparent swath.

original popup start point, and the opposing side drawn across the widest diagonal of the CoR²D item (that which produces the largest angle at the originating point). Transparency is used to prevent occlusion.

Multiple CoR²D Placement

In order to provide multiple operations for a particular operand, it is necessary to simultaneously display several CoR²Ds for a single object. Upon activation, the CoR²Ds are initially placed in a circle around the originating point. Once activated, the CoR²Ds can be moved and re-oriented by the users. We describe two separate implementations of CoR²Ds that handle user interaction differently: grouped CoR²Ds and independent CoR²Ds.

Grouped CoR²Ds

Figure 3 shows an invoked grouped CoR²D while Figure 4 shows the state transitions for grouped CoR²Ds. Our technique implements a handle for a group or cluster of CoR²Ds so that the CoR²Ds can be moved, reoriented and dragged across a distance on the large display in unison. It is important to notice that this handle is displayed immediately when the CoR²Ds are invoked, thus in one motion without lifting the inputting hand or device, it is then possible to move the CoR²D to a desired location on the display. The direction of the movement is used by our implementation as follows. On a horizontal surface, the direction of movement can indicate the position of the user or the position of a receiving user, when the CoR²Ds are passed to another side of the tabletop, and thus can help to decide what the correct orientation of the CoR²D is. This is especially important for text-based CoR²Ds. Therefore, our implementation of CoR²Ds keeps a record of the original starting screen location of the CoR²D, as well as its current position as the user is using the CoR²D. When the user selects the handle, the drag motion changes the current position and orientation of all CoR²D items at the same time. Once the CoR²Ds have been moved to and placed at the desired screen location, an action can be carried out by selecting the appropriate CoR²D tool item or menu item. The selection of a CoR²D item is used also as the dismissal mechanism.

Independent CoR²Ds

Figure 5 shows the state transitions for the second implementation. Here, many independent CoR²Ds (as

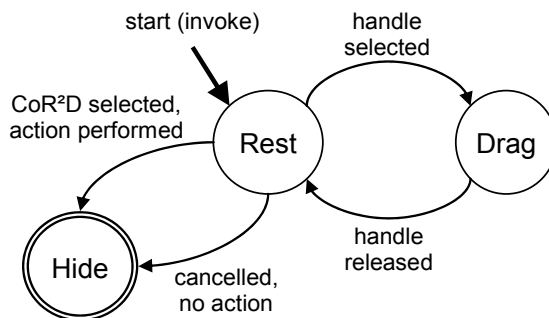


Figure 4. State diagram for grouped CoR²Ds.

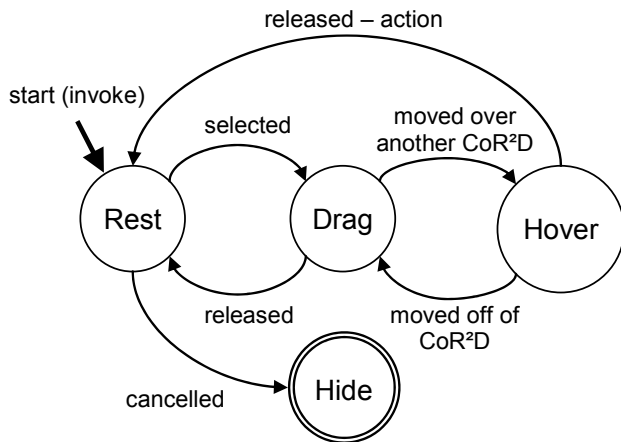


Figure 5. State diagram for independent CoR²Ds.

illustrated in Figure 2) originate from the point of activation of the same source object. Each CoR²D can be translated and rotated independently by the user to a desirable location and suitable orientation. A CoR²D can then be dragged on top of another CoR²D or object to invoke an action. When two CoR²Ds are involved, each can be moved in parallel, either by two different hands or two different users, to complete the interaction. Independent CoR²Ds support persistent tools and multiple activations of tools. Therefore, the dismissal of independent CoR²Ds requires a separate interaction. For independent CoR²Ds, a double-tap is used both for activation and dismissal.

INITIAL EXPERIENCE WITH COR²DS

CoR²Ds have been used in a few tabletop applications. In one application, CoR²Ds were used to implement popup meta-data for a multi-user tabletop interface that supports collaboration among bio-diversity researchers. One to four researchers sit around the table to browse, label, and search through digital photographs of flora and fauna taken during field expeditions. Another application was a collaborative park design workbench in which miniature physical objects (such as benches, trees or pictures of botanical items) are represented on a co-design tabletop. CoR²Ds are used to bring up details of design artifacts to be shared among three designers. A last application used multiple independent CoR²Ds to implement tools for collaborative image manipulation. The feedback so far has been positive – users like the ability to be able to freely drag a persistent tool set across the table, sharing with others while maintaining the visual knowledge of which document the menu or tool comes from. One problem identified is the activation methods. Currently, double-tap is used as the activation interaction for CoR²Ds. This works well on some displays, and is problematic on others. We are in the process of studying and experimenting to discover the most appropriate interaction mechanisms for the activation and dismissal of CoR²Ds on direct touch tabletops.

CONCLUSION & FUTURE WORK

CoR²Ds are a new popup widget for multi-user large-screen horizontal surfaces, designed to address the problems of

occlusion, readability and reaching, providing context on a cluttered display, and allowing concurrent interaction. We have implemented a set of proof-of-concept CoR²Ds on a DiamondTouch [3] tabletop and presented our initial usage experience. We are refining the design of CoR²Ds through user evaluation and exploration of application development.

ACKNOWLEDGMENTS

We would like to thank Merrie Ringel Morris, Yvonne Rogers and Richie Hazlewood for discussion on the application of CoR²Ds, and Mike Wu for implementing the activation method for grouped CoR²Ds.

REFERENCES

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. & Zierlinger, A. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. In *Proc. Interact 2003*, IOS Press (2003), 57-64.
2. Bier, E. A., Stone, M., Pier, K., Buxton, W. & DeRose, T. Toolglass and magic lenses: The see-through interface. In *Proc. SIGGRAPH 1993*, ACM Press (1993), 73-80.
3. Dietz, P. & Leigh, D. DiamondTouch: A multi-user touch technology. In *Proc. UIST 2001*, ACM Press (2001), 219-226.
4. Fitzmaurice, G., Khan, A., Pieké, R., Buxton, W. & Kurtenbach, G. Tracking menus. In *Proc. UIST 2003*, ACM Press (2003), 71-79.
5. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., & Kurtenbach, G. A remote control interface for large displays. In *Proc. UIST 2004*, ACM Press (2004), 127-136.
6. Morris, M. R., Ryall, K., Shen, C., Forlines, C., & Vernier, F. Beyond “social protocols”: Multi-user coordination policies for co-located groupware. In *Proc. CSCW 2004*, ACM Press (2004), 262-265.
7. Morris, M. R., Morris, D., & Winograd, T. Individual audio channels with single display groupware: Effects on communication and task strategy. In *Proc. CSCW 2004*, ACM Press (2004), 242-251.
8. Rogers, Y., Hazlewood, W., Bleviss, E., Lim, Y-K. Finger talk: Collaborative decision-making using talk and fingertip interaction around a tabletop display. *Ext. Abs. CHI 2004*, ACM Press (2004), 1271-1274.
9. Shen, C., Everitt, K. M., Ryall, K. UbiTable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In *Proc. UbiComp 2003*, (2003), 281-288.
10. Shen, C., Lesh, N. & Vernier, F. Personal digital historian: Story sharing around the table. *ACM Interactions*, 10 (2), ACM Press (2003), 15-22.
11. Shen, C., Vernier, F. D., Forlines, C. & Ringel, M. DiamondSpin: An extensible toolkit for around-the-table interaction, In *Proc. CHI*, ACM Press (2004), 167-174.