Dynamic Portals: A Lightweight Metaphor for Fast Object Transfer on Interactive Surfaces

Simon Voelker   *Malte Weiss*   *Chat Wacharamanotham*   *Jan Borchers*

RWTH Aachen University
52056 Aachen, Germany
{voelker, weiss, chat, borchers}@cs.rwth-aachen.de

ABSTRACT

We introduce Dynamic Portals, a lightweight interaction technique to transfer virtual objects across tabletops. They maintain the spatial coherence of objects and inherently align them to the recipients' workspace. Furthermore, they allow the exchange of digital documents among multiple users. A remote view enables users to align their objects at the target location. This paper explores the interaction technique and shows how our concept can also be applied as zoomable viewport and shared workspace.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces.—Interaction styles.

General terms: Design, Human Factors

Keywords: Interactive Tabletops, Orientation, Spatial Arrangement, Remote View, Portals

INTRODUCTION

Interactive tabletops are promising devices for enriching collaborative work in everyday tasks. One of their key advantages over desktop computers is their ability to join the flexibility of digital documents with the benefits of regular tables. They can be approached from all sides and enable the full spectrum of social protocols which benefits collocated collaboration. Beyond that, multi-touch input in combination with physical devices and tangibles provides a manifold and efficient way to interact with digital data.

The non-physical nature of digital documents on tabletops hampers their exchange among users. To achieve an efficient work flow, users need quick techniques to hand digital documents to other users. Furthermore, a recipient should not be burdened with realigning the document to read it. Beyond that, virtual documents can be composed of multiple elements, such as a mind map. This structure is a crucial part of the information and must be maintained when transferring. Much research has been done to aid either preserving spatial information [6, 8] or automatically aligning objects to
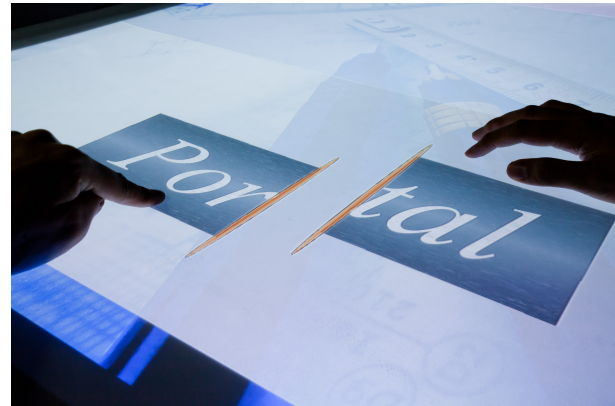
Figure 1: Dynamic Portals is a light-weight technique to exchange digital documents on interactive surfaces.

be readable by viewers [3, 9]. This results in multiple methods and visual widgets, each of which need to be learned.

In this paper, we present Dynamic Portals, a lightweight interaction technique that allows users to exchange objects while preserving their spatial arrangement (Fig. 1). Furthermore, objects are inherently aligned to the recipient's workspace. Our technique involves a low interaction overhead, making it suitable for frequent use on tabletops. Finally, our approach is easy to implement and can also be used as zoomable viewport and shared workspace.

RELATED WORK

Many techniques intend to determine proper orientations of objects that are moved across the table. They are either *location-based*, using the objects' positions on the table, or *interaction-based*, interpreting how users move or transform these objects. [9] presents an automatic method that aligns the document either towards the center or towards each collaborator. A semi-automatic approach allows users to define object orientation of different areas on the surface [3]. Rotate'N Translate, an object-based technique, couples orientation to translation using a virtual friction against the direction of movement [5]. While space-based approaches do not require users to orient each individual objects, it can complicate interaction when users need some objects to be manually oriented.

To preserve spatial arrangement grouping commands or widgets, e.g., storage bins [8] or TableTrays [6] were proposed.

Although these techniques map directly to physical actions, they add overhead from group/ungroup or add/remove actions. In addition, they do not preserve spatial layout of objects.

To surpass the distance limits either on the same or between different surfaces indirect manipulation methods, e.g., Vacuum [2] or I-Grabber [1], provide widgets for users to extend their reach to manipulate distant objects. While these methods raise collaborators' awareness during manipulation, they clutter the shared surface of the table. Other methods, e.g. [4, 7, 10, 11] let user directly interact with objects on local or remote space directly through a virtual window. This metaphor can be extended to allow object transfer and remote collaboration.

Dynamic Portals extend the portal metaphor to assist preserving spatial orientation and arrangement on tabletop interaction. Despite the extensions of features, it simplifies interactions with less visual demand and less interaction steps and allows emerging interaction from the combination of previously separated concerns on spatial information on tabletops.

DESIGN CONSIDERATIONS

Our interaction technique aims to augment tabletop tasks without interfering with them. From this goal we derived a set of requirements that a transfer technique should comply with:

- *Lightweight:* An object transfer should be ad-hoc and fast, without the need for mental context switches. Accordingly, the transfer technique involved should be intuitive and easy to trigger.
- *Unobtrusive:* The technique should consume a minimum amount of real estate on the table. Furthermore, an object transfer should not disturb other interactions on the table (which occur, e.g., when an object is flung across a workspace).
- *Maintain spatial coherence:* Some tasks involve the spatial arrangement of multiple objects. If they are transferred, the technique should maintain the spatial relationship between all objects.

DYNAMIC PORTALS

To fulfill the aforementioned properties, we developed the so-called *Dynamic Portal*, inspired by the computer game Portal[1]. Conceptually, a Dynamic Portal is a thin line in the tabletop interface that transfers virtual objects crossing that line to distant portals.

Fig. 2a illustrates the formal definition of a portal. A portal consists of two points P and Q that yield an up vector $u = Q - P = (u_x, u_y)^\top$. Every portal has a direction vector $d = (u_y, -u_x)^\top$. The two orthogonal vectors span a local coordinate system. The line $\overline{PQ}$ separates the space around the portal into two half-spaces A and B.

Portals are displayed as a narrow ellipse (Fig. 2b). A faint glow indicates the orientation of the portal. Portals that are linked to each other have the same color. We intentionally
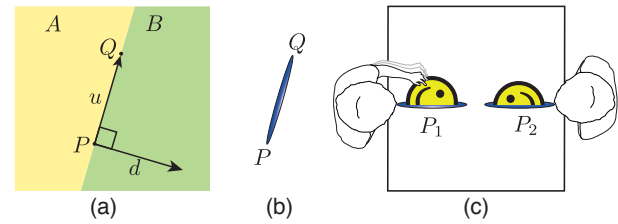
Figure 2: Definition of portal (a) visualization (b) and orientation of objects being sent through a portal (c)

did not visualize links between portals any further to avoid interference with the shared working area.

To transfer an object, the user can drag the object through the portal from either side. When an object is moved from a portal P_1 to P_2 it is mapped from the coordinate system of P_1 to the coordinate system of P_2. Furthermore, it appears at the opposite side of the target portal, i.e., an object being pushed from the A side of a portal appears at the B side of the target portal. This interaction maintains *orientation* of transferred objects as perceived by the recipient, e.g., when users are sitting face to face (Fig. 2c).

INTERACTION DESIGN
Opening and Closing Portals

A portal is easily created, or *opened*, by dragging a straight line on the surface. To avoid ambiguous input, the portal is only created if this line starts and ends in an area which is not covered by any other virtual object. The dragging direction determines the orientation of the portal; start and end position of the line correspond to P and Q in the previous section respectively. The direction should be consistent among users. Thus, we decided that a user should open portals by dragging a line towards herself. If a portal is not needed anymore, it can be removed, or *closed*, by pinching its end points together.

Linking and Unlinking Portals

A portal can either be *unlinked*, *linked to another portal* (1:1 mapping), or *linked to a set of portals* forming a complete graph, i.e., an object being sent from any portal in the set will appear in all other portals in this set. Because linking is frequent, we designed two methods that are optimized for different scenarios: (1) each of two successively created portals are automatically linked, (2) users can change portal link using a directional quasi-mode gesture.

The first method is designed for quick and easy use in a single user scenario. For example, imagine a user who wants to store her current work progress in a different area. Two line gestures, one drawn next to the data and the other within a dedicated storage area, yield two linked portals. After sending the objects, the two portals can be simply closed as described before. Note that if the user creates two more portals, the third portal would be linked to the fourth one, without any association with the first pair.

The second method is designed to support two common scenarios: linking to a portal that is out of reach and linking a portal to a set of portals. If the user holds down her finger

on a portal, a directional pie menu appears (Fig. 3a). Each pie points to a distant portal which the selected one could be linked to. To link, users drag towards the direction of a desired portal until the respective item is highlighted. After releasing the finger, both portals are linked. The link can be easily removed by repeating that action. The user can link a portal to a set of linked portals by linking it to any portal in that set. This allows distributing objects to multiple targets.

We chose this quasi-mode gesture because it does not require any additional persistent controls in the interface. Furthermore, the gesture uses direction information existing on the table, making it more intuitive than on-screen buttons. If several portals are opened in the same direction, a hierarchical pie menu can be used to solve potential ambiguities.

In a collocated multi-user context, the aforementioned gesture allows users to link to other users' portals without an explicit request. A more strict variant of our quasi-mode could require two users to perform the gesture towards each other, creating a bilateral agreement. However, we kept our simple solution as we believe that collocated users conform with social protocols when exchanging data.

Sending Objects
A user can easily send an object by dragging it across a portal. As soon as a part of the object is dipped into a portal, that part directly appears at all linked portals, in their respective coordinate systems. As mentioned above, objects are always sent to the opposite side of the destination portals, i.e., an object is dipped into side A of a portal will exit at side B of all target portals. Since the sent object is aligned to the recipient's local coordinate system no further rotation is required. The sender can always drag an object out of a portal as long as it has not completely disappeared from the sender's side.

If multiple objects have to be sent, they could be dragged into portals one after another. However, this would dissolve the spatial relationship of these objects. In graphical applications, e.g., a card-sorting program or an architectural tool for aligning furniture in a room, this would destroy the crucial spatially connected information. We solved this issue of maintaining spatial coherence by introducing a *scanning* gesture. Rather than subsequently moving the objects through the portal, the portal itself can be dragged and swept across those objects. As long as the portal is not released, every sent object re-aligns all previously passed objects to maintain their original spatial alignment in the new local coordinate system (Fig. 3b). This gesture is easy to learn and avoids mode switches, such as grouping the objects before transferring them. If desired, the user can also skip particular objects by not including them into the sweep process; the portal still maintains the original spatial relationship between all transferred objects (Fig. 3c).

Scaling Objects
Since portals transform objects to the local coordinate system of the receiving portals, they can inherently be used to scale objects. If the target portal is smaller than the source one, the object is scaled down; if the receiving portal is larger, the object is enlarged (Fig. 3d). Using this technique, users can scale up objects they want to edit, or minimize objects
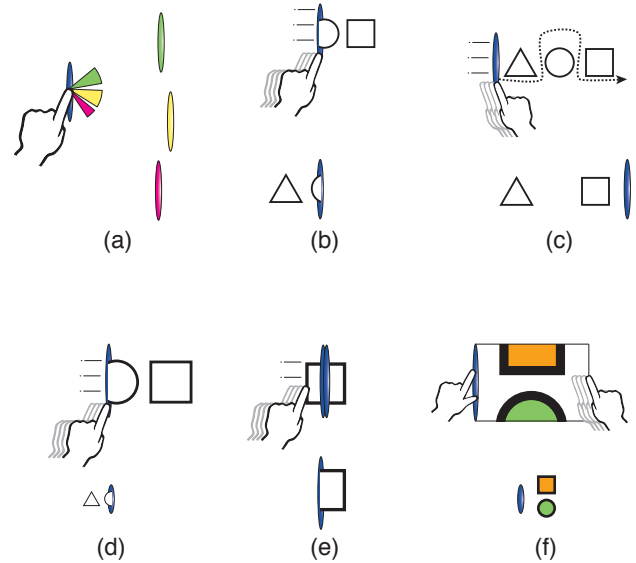

Figure 3: Interacting with portals: (a) Linking portals with a pie menu. (b) Scanning multiple objects preserves their spatial relationship. (c) Some objects, e.g., a circle here, can be skipped. (d) The size of objects at the destination is determined by portal size. (e) Duplicating an object. (f) Remote view as a detail viewport

to store them for later use. The scaling maintains the ratio between objects' and portal's height at the source and the target. To avoid unwanted resizing operations, scaling is only performed if the factor exceeds a certain threshold.

Duplicating Objects
In some scenarios it can be helpful to retain a copy of the object being sent. Imagine a meeting in which a moderator distributes the agenda document to her colleagues. As usual, she opens a portal to all participants. In order to keep an own version of the agenda, she double-taps the portal. It is now switched to *duplication mode*, which is illustrated by a double ellipse (Fig. 3e). If the moderator now drags a file across the portal line, copies are sent to the destinations, but the original passes the portal without vanishing. Double-tapping the portal again toggles it back to the default mode. A user can also directly open a portal in duplication with a two-finger gesture for drawing two parallel lines in free space.

Remote View
In a collocated setting, users can normally see how transferred objects appear at the destination. However, on large surfaces or when linking a portal to a remote location, they can barely see the effect of the sent objects. The missing feedback may lead to significant cluttering at the target portal, e.g., when multiple users sent objects to the same shared space; the receiving user is then required to reorganize all received data. To lessen this workload, we introduce a *remote view* for portals.

Imagine a brainstorming session. Alice assembles the results in a mind map. She has also opened an adjacent portal to let her colleagues participate in this process. Bob types an

important keyword that he wants to add to the mind map. Rather than just sending that to Alice's portal, he holds down two fingers on his portal and unrolls a *remote view* by dragging a third finger away from the portal (Fig. 3f). This view displays the workspace at the destination portal. All interactions within this view are interpreted as if the user was interacting with the surface on the destination side. Bob expands this view by dragging its border until Alice's mind map is entirely visible. He can now move his keyword across the portal line into the remote view and arrange it *at the target location* such that it fits into the mind map. In contrast to Frisbee [4] the destination portal does not overlap the mind map which allows Alice to continue her work on the mind map. After that he closes the remote view by dragging the border back to the portal line. His keyword is now incorporated into the mind map in front of Alice. Note that a remote view is only available if a portal is linked to a single other portal.

The same technique can be used to create *overview-detail viewports* on the tabletops similar to [4, 10]. Imagine a user had minimized some of his documents but wants to continue working with them now. Rather than scaling them up again, she can create a small portal next to the documents and a large one in front of himself. If she now expands a remote view on the large portal, she implicitly opens an upscaled *workspace* of the minimized documents and can directly interact with them. Note again that all touches in the remote view are interpreted as if they were done in the area with the minimized documents. Thus, using this technique users can precisely interact with very small objects without enlarging them. By changing the size of the involved portals, the user can change the scale factor on the fly. Another example is an illustrator that employs a remote view to modify a detail of his work while still seeing the overall view. In the same spirit, remote views can be used for indirect interaction, e.g., to control a wall display from a tabletop device.

Finally, remote views can be combined to shared workspaces. Two users sitting vis-à-vis can open a portal in the center of their workspace. If one user unreels a remote view to the left of his portal, and the other user creates one to the right, they now share a workspace where input and output is equal for both users. Although the space is shared, the ownership of each side of shared space is always visible because the local and remote side have different backgrounds.

CONCLUSION AND FUTURE WORK

Dynamic Portals is a simple metaphor for transferring virtual objects to distant positions. It can be used ad hoc and maintains the spatial coherence of the objects being transferred. Remote views support the integration of sent documents into the target space and can be used as remote workspaces.

In future work, we want to conduct user studies to explore Dynamic Portals in the multi-user context. Our current concept assumes that users follow social protocols when collaborating on a table. However, it might be useful to implement authorization and privacy protocols, e.g., to avoid distant users to see or manipulate one's workspace without permission. We are, furthermore, interested in how well our concept performs if many users are involved in the interac-

tion. Finally, we want to investigate cross-device interaction in more detail, i.e., sending objects from one interactive surface to another, or even to remote users over the internet.

REFERENCES

1. M. Abednego, J. Lee, W. Moon, and J. Park. I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *Proc. ITS '09*, 61–64. 2009.

2. A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. In *Proc. CHI '05*, 361–370. 2005.

3. P. Dragicevic and Y. Shi. Visualizing and manipulating automatic document orientation methods using vector fields. In *Proc. ITS '09*, 65–68. 2009.

4. A. Khan, G. Fitzmaurice, D. Almeida, N. Burtnyk, and G. Kurtenbach. A remote control interface for large displays. In *Proc. UIST '04*, 127–136. 2004.

5. R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *Proc. CHI '05*, 601–610. 2005.

6. D. Pinelle, T. Stach, and C. Gutwin. TableTrays: temporary, reconfigurable work surfaces for tabletop groupware. In *Proc. TABLETOP '08*, 41–48. 2008.

7. D. Pinelle, T. Stach, D. Stuckel, M. Nacenta, J. Dyck, and C. Gutwin. Cutouts: A flexible interaction infrastructure for tabletop groupware. In *Proc. CSCW EA '06*. 2006.

8. S. D. Scott, M. S. T. Carpendale, and S. Habelski. Storage bins: Mobile storage for collaborative tabletop displays. *IEEE COMPUT GRAPH*, 25(4):58–65, 2005.

9. F. Vernier, N. Lesh, and C. Shen. Visualization techniques for circular tabletop interfaces. In *Proc. AVI '02*, 257–265. 2002.

10. S. Voida, M. Tobiasz, J. Stromer, P. Isenberg, and S. Carpendale. Getting practical with interactive tabletop displays. In *Proc. ITS '09*, 109–116. 2009.

11. D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Table-centric interactive spaces for real-time collaboration. In *Proc. AVI '06*, 103–107. 2006.