Chapter 10
Supporting Atomic User Actions on the Table

Dzmitry Aliakseyeu, Sriram Subramanian, and Jason Alexander

Abstract One of the biggest obstacles that application developers and designers face is a lack of understanding of how to support basic/atomic user interactions. User actions, such as pointing, selecting, scrolling and menu navigation, are often taken for granted in desktop GUI interactions, but have no equivalent interaction techniques in tabletop systems. In this chapter we present a review of the state-of-the-art in interaction techniques for selecting, pointing, rotating, and scrolling. We, first, identify and classify existing techniques, then summarize user studies that were performed with these techniques, and finally identify and formulate design guidelines based on the solutions found.

Introduction

Research into tabletop systems began more than 15 years ago with the DigitalDesk [1], which proposed the idea of a horizontal computationally enhanced interactive surface (aka as a digital table or a tabletop system). Since then, a large portion of research prototypes have aimed at being point-designs (or one-off designs) that highlight the feasibility of the design or promote new opportunities for tabletop systems. In doing so, they also propose different kinds of interaction techniques; for example, The InteractiveDESK [2], introduced the concept of linking physical objects, such as a scrapbook, to digital information, like a computer folder; the use of a three-dimensional space above the table was first proposed within the Responsive Workbench [3]. Active Desk [4] and metaDESK [5] were the first systems to implement the concept of using physical objects to both control and represent digital information.

D. Aliakseyeu (✉)
User Experiences Group, Philips Research Europe, 5656AE Eindhoven, The Netherlands
e-mail: dzmitry.aliakseyeu@philips.com

C. Müller-Tomfelde (ed.), *Tabletops – Horizontal Interactive Displays*,
Human-Computer Interaction Series, DOI 10.1007/978-1-84996-113-4_10,

In recent years there has been a growing interest in bringing these various tabletop systems to the market. One of the biggest obstacles that application developers and designers face is a lack of understanding of how to support basic/atomic user interactions. User actions, such as pointing, selecting, scrolling and menu navigation, are often taken for granted in desktop GUI interactions, but have no equivalent interaction techniques in tabletop systems. Recognizing this need, the research community has recently started exploring various techniques to support atomic interactions in different tabletop scenarios. For example, Aliakseyeu et al. [6] proposed and evaluated mechanisms to scroll documents on pen-based tabletop systems while Benko et al. [7] looked at virtual objects selection mechanisms.

Most of these basic interactions are standard for conventional GUI based systems, and have been extensively studied. However, the adaptation of these atomic user actions to the tabletop environment is not straightforward:

- First, tabletop systems offer richer interaction possibilities, therefore simple translation from the standard GUI is unlikely to be optimal;
- Second, the size and horizontal position of the display introduces new challenges such as large reaching distances and range, display occlusion by users' hands and physical objects, lack of precision, the need to orientate objects, the need for sharing, and the lack of tactile feedback.

This chapter offers a structured review of research efforts in supporting atomic user actions. We provide a survey of interaction solutions for tabletop systems and a classification that aims to help designers choose suitable implementations of interaction techniques. We, first, identify and classify existing techniques, then summarize user studies that were performed with these techniques, and finally identify and formulate design guidelines based on the solutions found. When formulating design guidelines, the main measures that we take into account are performance, error rate, user preferences, practical issues and compatibility with other solutions. Later chapters in the book look at other performance measures such as coordination and collaboration.

To keep the chapter focused we only explore selecting, pointing, rotating and scrolling.

Atomic Actions and Chapter Layout

We define the chosen atomic actions as follows:

- *Selecting*. Selection is the process of highlighting a target object on the screen. Examples of this include touching the object with the hand (finger), stylus or physical object and clicking a button on a stylus, physical object or other interaction device.

- *Pointing*. Pointing is the process of moving a cursor from an initial position to a target object on the screen. The cursor reacts to, for example, the hand (finger), stylus, physical object or other interaction device movements. The literature documents four broad categories of pointing techniques: deposit, retrieve, local-operate, distant-operate and combinations of the above.
- *Rotating*. Rotating is the ability to reposition (translate) and reorient (rotate) physical objects, such as printed documents, photos and charts. Rotation forms a vital part of collaboration in tabletop settings. Digital tables offer the potential benefit of bringing together the traditional face-to-face setting with advantages of an electronic information processing unit. Reorienting documents is one such interaction that is important for digital tables but is generally implicit in the screen's orientation in traditional desktop interaction.
- *Scrolling & Panning*. Scrolling is an important interaction technique for supporting many daily tasks within large information spaces. Scrolling actions shift the viewport to content that is of interest and that currently resides off-screen. Usually 1D shifting (scrolling a text document) is referred to as *scrolling,* while 2D shifting (exploring a map) as *panning*. For consistency, in this chapter we will only use the term *scrolling*, and if necessary we will clarify if it is 1D scrolling or 2D scrolling.

Tabletop systems support a large variety of input mechanisms. The choice of a particular input will define how the different atomic actions are performed. There are three common input mechanisms for interacting with a tabletop system:

- Finger(s), and/or hand(s) and other parts of the fore-arms (DigitalDesk [1], DiamondTouch [8], Microsoft Surface ©, Entertaible ©);
- Stylus (pen) (VIP [9, 10]);
- Specific tangible objects, such as game pawns, tool representations, etc. (BUILD-IT [11], metaDESK [5], Microsoft surface ©, Entertaible ©).

Every input method has its pros and cons; for example, selection of an object by simply touching it with your finger is a very intuitive technique used in touch screens, however issues such as occlusion, parallax and reach distance may lead to poor performance of this otherwise simple and straightforward technique. These issues are common among many interaction techniques and can prevent reliable interaction.

Usually, a particular interaction technique addresses one or more of these issues. These issues are highlighted as we discuss each atomic action.

The next four sections describe our four selected atomic actions. Each section is laid out as follows: first, we introduce the atomic action, second, we describe the most relevant existing techniques that perform this action, third, we outline relevant user studies and fourth we provide design recommendations and open questions for the atomic action.

Selecting

We consider selection as a distinct operation from pointing. However, the performance with target selection is commonly studied with pointing and selection as one combined operation [12, 13]. Unfortunately, these two elements are rarely investigated in isolation. However, some evidence suggests that selection alone (i.e. button clicking) without pointing can consume a significant amount of the total target selection time [14, 15]. As a result, enhancing the selection mechanisms on an input device can lead to more efficient interactions. Furthermore, selection techniques differ in their intuitiveness to users, and the accuracy that users can achieve.

Selection Techniques

This section provides an overview of existing selection techniques. Not all techniques that are discussed were specifically designed for tabletop systems; however, many are included as they could equally be applied in a tabletop situation (some techniques were developed for large vertical displays, some for portable pen and touch based devices, and some for desktop systems).

The two most prominent concerns for selection are *occlusion* and *two state input*. Most touch based tabletop systems are two-state devices (Fig. 10.1): (0) out of range and (1) tracking. Conversely, the mouse and some stylus inputs are three-state devices: they additionally support a (2) dragging state when the mouse button or pen-tip is pressed down. This means that pointing and selection may interfere with each other if no measures are taken, as both use the tracking state to initiate the operation [16]. The two state input is therefore is not a concern for selecting "non-movable" objects such as those used for menu selections and text input.

Most of the finger/hand based techniques either address the two-state issue prevalent on touch screen or the issue of covering small targets. Some techniques address both issues.

The **two state** issue is addressed by the following techniques: *Take-off* [17], selection on finger down-up transition within target area; *Tactile* (also referred as

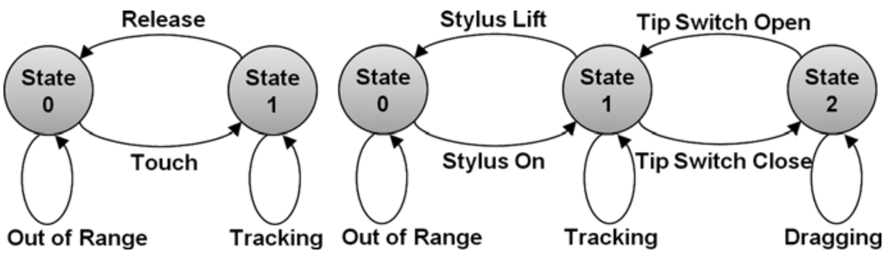

Fig. 10.1 Two state model as applicable in basic interactive tabletop systems (*left*) and three state model as applicable in various stylus based systems [16], such as from Wacom © (*right*)

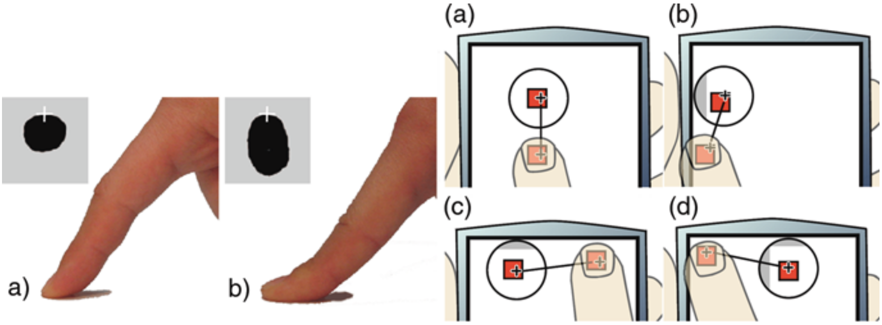

Fig. 10.2 (*Left*) The SimPress press-through technique as implemented by Benko et al. [7]; (*Right*) Shift technique [23]

press-through) [18], selection on changing from low to high finger pressure within target area; *SimPress* [7] a special variant of the press-through technique (Fig. 10.2, left); *Double tap* [19], selection on finger down-up-down-up (tap-tap) within target area; *HoTTop* click gesture [20], selection is done by placing two fingers on the area of interest (document page); *DTMouse* [21]: selection or mode change performed by tapping with the second finger; *SmartSkin mouse press* [22]: selection on hand-surface distance, where a distance between the surface and the palm is used to distinguish between pressed and released states.

In all of the solutions discussed above, the finger covers (part of) the target object, which can lead to issues when selecting small objects. Widgor et al. [24] showed that selection precision can be learned and improved with practice. However, when the user requires visual feedback (e.g. during text selection) and when the user is selecting one object among multiple small closely located objects, alternative solutions might be needed.

The issue of covering is addressed either by creating a cursor offset or by local zooming. Although some of these techniques were developed and evaluated on PDAs only, which have relatively small screens and target objects, this potentially is also an issue for tabletop applications. Offset techniques include: *cursor offset* [7] that places a cursor above the finger, enabling it to select small targets without covering them; *dual finger offset* [7] that places a cursor above the finger, with the offset triggered by placing a second finger; *shift* [23] that creates a callout near the finger with the content that is under the finger (Fig. 10.2, right); *dual finger stretch* [7] that stretches the area of interest, making the selection of small objects simpler; *under the table selection* [25] where the hand of a user is placed under the table, therefore it does not occlude the targets that are projected on top of the table; *rub-pointing* [26], combines zooming and selecting into one movement; *zoom-tapping* [26], employs tapping with the secondary finger for zooming.

With a stylus, users commonly invoke a selection by directly tapping over an object (quickly touching and then releasing the stylus without horizontal movement, similar to a button click). Since tapping does not reflect how people naturally use

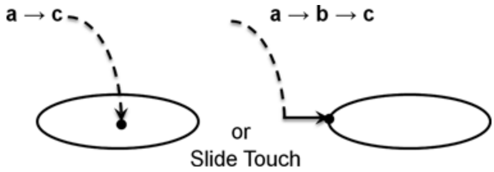

Fig. 10.3 Slide touch [27]

notepads, where writing and making checkmarks is common, designers have developed an alternative referred to as *touching* [27]. Unlike tapping, which requires that a pen touch a screen and be lifted directly over the target to select it, touch interactions only require that the target be touched at some point (Fig. 10.3). As a result, touching supports the selection of targets by crossing them, making checkmarks and even tapping. An example of such a technique is a *slide touch* [27]. With this technique, the target is selected when the pen touches it for the first time – the pen can initially either land on the target or outside of the target (Fig. 10.3).

Results show [15, 27] that touching is a viable alternative to tapping for completing selection, even for the elderly [15]. Other pen-based systems have shown to provide effective selection mechanisms by crossing targets. With *CrossY* [28] the pointing is eliminated and instead selection happens in one fluid motion by crossing an object.

Various stylus based systems use a three state model for selection in a similar manner to the conventional desktop mouse. For example, to select an object with the Wacom© digital pen, touching the object is not enough, the pen must also be pressed against the surface so that the tip switch is activated. This allows pointing and selecting to be differentiated.

Tangible objects require slightly different interactions for selection. Tangible objects must usually be placed on top of a virtual object for selection to take place. The tracking technology will determine the type of mode switch employed for selection to take place. For example, the BUILD-IT system [11] uses tangibles that are active as soon as they are placed on the table. To select a virtual object the tangible (physical object) must be placed on top of the object and to deselect it the tangible needs to be covered by the hand. The VIP system [29], which is also vision based, uses a modified approach: to select or deselect the object, only a specific part of the tangible needs to be covered (leading to a "virtual button").

Comparison of Selection Techniques

Potter et al. [17] compared the following three selection techniques: First-contact, land-on, and take-off. They concluded that *take-off* has the best performance in terms of speed and error rate and it was most preferred by the users.

Benko et al. [7] studied the press-through technique without tactile feedback on a touch screen (SimPress). The authors concluded that SimPress worked only if the user was always approaching the tabletop from the same direction; otherwise the orientation of the hand and arm had to be taken into account.

Ren and Moriya [27] compared six selection strategies for pens. The comparison used a standard Fitts law experiment: participants selected a number of appearing targets as fast and as accurately as possible using different selection strategies. They found that Slide Touch (Fig. 10.3) is the best technique in terms of selection time, error rate, and subjective preference.

Hourcade et al. [15] compared a touching techniques and tap selection techniques. The touch technique was similar to the Slide Touch techniques proposed by Ren and Moria [27]. The study focused on accuracy and age differences. They found that all age groups (18–22, 50–64, 65–84) were more accurate when touching than when tapping circular targets with a diameter of 16 pixels = 3.84 mm.

Design Recommendations and Open Questions

When selecting objects that require only two-sate input (e.g. menus and text input):

- Touch-based techniques should be used when selecting objects. Examples of these techniques include *slide touch* [27], *take-off* [17] and *direct touch*. *Take-off* or *direct touch* should be employed when there is a high density of potential targets.
- For selecting small objects *cursor offset* can be used; *cursor offset* can also be used to remedy the input-output parallax. However, if possible, *cursor offset* based techniques should be avoided since it was shown that they require significantly more time than simple touch techniques and the system needs to know the orientation of the user (which can be either calculated by analyzing the finger surface contact area, or avoided by using midpoint between two fingers) [7]. The choice for cursor offset should be based on a desired performance and error rate.
- As an alternative to *cursor offset* [7] and for selecting very small targets, local zooming (e.g. d*ual finger stretch* [7]), or showing a copy of the occluded screen area in a non-occluded location (e.g. s*hift* [23]) can be employed. These techniques are the most appropriate both in terms of users' performance and preference.

The corresponding sizes for "small" and "large" targets need to be experimentally determined, however we can safely say that targets larger than the finger tip can be considered large, and targets that are smaller than the input resolution can be considered small.

When selecting objects that require three-state input (e.g. dragging and activating):

- *Bimanual or multi-finger* techniques are the most effective. The object can be selected using one of the techniques described above and then the second hand

or another finger can be used to change the object's state. Different users can be discriminated by using a threshold distance between the detected finger positions. Alternatively, single point techniques such as *SimPress* [7] can be employed.
- A three state system (mode switch) can also be implemented using tangible objects. For example, an interactive physical object that is used for both selection and positioning could communicate button presses on the object to the table.

There are still a number of questions that remain unanswered:

- *Universal selection method*. Based on the literature review we can conclude that no single technique can address all aspects of selecting on the tabletop, therefore a hybrid (mixed) technique with several modes is necessary. It is however unclear what combination would be the most efficient, as this is likely to be dependent on the task and context.
- *Three state-input (mode switching)*. Direct touch is the simplest and fastest technique for selection; however, it does not immediately support mode switching. Techniques that support mode switching suffer from either lower performance or a lack of robustness. The problem of mode switching can be addressed both through software solutions (SimPress) or hardware (for example, by making tabletop surfaces pressure sensitive). The universally best solution still requires further research.
- *Occlusion*. Occlusion is addressed by a large number of techniques. However, Widgor et al. [24] found that in many situations occlusion is not a problem, especially if the target size is large. However, as the input and output resolution of tabletop systems continues to grow, occlusion may become a main concern when choosing the most suitable selection technique.

Pointing

Pointing is the process of moving a cursor from an initial position to a target object on the screen, the cursor reacting to the hand (finger), stylus, physical object or other interaction device movements.

The literature documents four broad categories of pointing: deposit, retrieve, local-operate, distant-operate and combinations of the above [30]:

- *Deposit*: These techniques allow users to move an object from within their vicinity to a distant location (e.g. Flick [19, 31]).
- *Retrieve*: These techniques bring distant objects closer to the user for selection (e.g. Drag-and-Pop [32]), but do not support relocation of the cursor to a distant location.
- *Local-operate*: These techniques allow the user to interact with objects that are within their hand-reach. They support cursor-offsetting and/or high-precision

control for local-control of the pointer (e.g. Dispositioned Cursor [33] and Dual-Finger MidPoint [7]).

- *Distant-operate*: These techniques allow users to relocate their cursor to a distant location to facilitating object manipulation in distant locations. Examples include the Perspective Cursor [34] and Push-and-Throw [35].
- *Combination techniques*: These techniques combine two or more of the functionalities described above. For example, Push-and-Pop [35] supports the retrieve and distant-operate functions and Radar Views [36] allow the user to deposit and retrieve.

Another important dimension of pointing techniques is the location of input [37]. The input location can be in the personal space (close to the user – within about one meter) or in the group space (the area between the users, visible to all of them but not always easy to access). Techniques that make use of personal input space are called *indirect* techniques; interaction with objects on the entire table is performed from the personal area. Techniques which make use of group input space are called *direct* techniques, interaction with objects happens directly on the place of the object itself. Mixed techniques are a midway between direct and indirect techniques and thus make use of input from the personal and group space.

Pointing Techniques

This section provides an overview of existing pointing techniques. Techniques will be presented based on the location of input (direct or indirect).

Direct Pointing Techniques

These techniques allow the user to interact with objects that are within their hand-reach. They often support cursor-offsetting and/or high-precision control for local-control of the pointer [37].

The most common techniques in this category are: *drag-and-drop,* an interaction technique where the user selects an object by touching it with a stylus, and deselects it by lifting the stylus from the tabletop [37]; *touch-input,* an interaction technique comparable to drag-and-drop – the difference is that a body part (e.g. a finger) is used instead of a stylus [38]; *pick-and-drop* [39], the user can "pick up" an object by touching it with a digital pen (or finger or any other suitable device), and then "drop" the object anywhere in the workspace by repeating the touch action in the desired location.

Unlike the techniques described above, the following techniques are coupled to a specific input mechanism (finger/hand touch or pen/stylus): *dispositioned cursor* [33]: shows the cursor just above the finger tip, combined with the take-off selection technique; *zoom-pointing* [33]: first the user zooms to a sub area defined by drawing a rectangle, they can then perform direct pointing at a finer scale, combined with the take-off selection technique; *cross-keys* [33]: uses four arrow keys around a visor to finely adjust the cursor, combined with the take-off selection technique; *precision-handle* [33]: uses a virtual pantograph for precise pointing, combined with the take-off selection technique; *dual finger offset* [7] is activated by placing a secondary finger anywhere on the surface, the cursor is subsequently offset above the primary finger by a predefined fixed amount (selection is done via the SimPress); *dual finger midpoint* [7]: positions the cursor exactly halfway between the two fingers, giving the user both a cursor offset as well as a variable reduction of cursor speed (selection is done via the SimPress); *dual finger stretch* [7]: adaptively scales the user interface (selection is done via the SimPress); *dual finger slider* [7]: the right finger (primary) controls the cursor, the left finger (secondary) invokes the invisible slider, with speed reductions modes achieved by moving the secondary finger towards the primary finger (selection is done via the SimPress); *SDMouse* [40]: emulates full mouse functionality through multi-finger input; *bubble cursor* [41]: is an improvement of area cursors [42] that allows selection of discrete target points by using Voronoi regions to associate empty space with nearby targets using hotspot; *DynaSpot* [43]: couples cursor activation area with it speed, the faster cursor is moved the larger the activation area of the cursor.

Indirect Pointing Techniques

Indirect techniques make use of the personal input space. They employ virtual embodiment and local and shared feedback [37].

Radar views [36] are an interaction technique that makes use of a miniature view of the entire workspace which is displayed in front of each user (Fig. 10.4); *bubble-radar* [44] combines radar views and bubble cursor techniques; *pantograph* (also referred as Push-and-Throw) [35] is an interaction technique in which the user moves the stylus in his/her personal space, as with radar, however, there is no workspace miniature; *telepointers* [37] are an interaction technique which is equivalent to the pantograph, but without the line that connects the stylus to the cursor; *drag-and-pop* [32] brings proxies of all potential targets in the direction of movement closer to the user (Fig. 10.5); *throw and flick* techniques [19, 31, 45, 46] use a simple stroke (with the pen or hand) on the table surface to slide an object, mimicking the action used to send physical objects across a table; *superflick* [47] adds an optional closed-loop control step to basic Flick; *HybridPointing* [48] allows the user to quickly switch between absolute and relative pointing thus combining direct and indirect pointing.

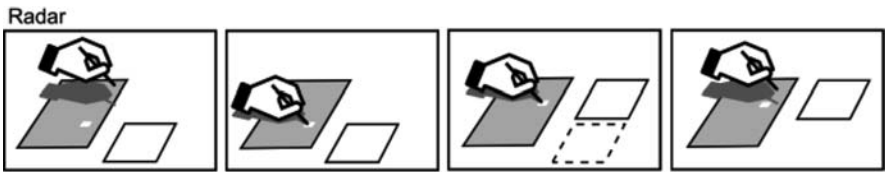

Fig. 10.4 Radar view interaction scenario [37]

Fig. 10.5 In drag-and-pop, each valid target icon in the direction of the drag motion creates a linked tip icon that approaches the dragged object. Dropping onto a tip icon saves mouse travel to distant targets [32]

Comparisons of Pointing Techniques

Albinsson and Zhai [33] compared the following high precision selection techniques using Fitts-law tasks: dispositioned cursor (take-off), zoom-pointing, cross-keys, and precision-handle. They concluded that take-off's one-step nature makes it fast when the target is large enough, but hard to operate accurately when aiming at single pixels; zoom pointing performed well on speed, errors, and user preference for all target sizes; cross-keys allowed the users to select small targets with low error rates.

Benko et al. [7] compared a number of dual finger pointing techniques that aimed to improve selection precision. They found that dual finger stretch performed the best on speed, error, and user preference measures. This was the only tested technique that did not provide a cursor offset.

Nacenta et al. [36] compared several pen-based pointing techniques like the Radar View, Pantograph and Pick-and-drop via Fitts-law tasks. Target width W was set at 80 mm and distance D at 25, 50, 80, 140 cm. They concluded that the Radar View was significantly faster than all other techniques in the ranges tested. Reetz et al. [47] compared Flick and Superflick with a Radar view for a variety of placement tasks. They found that flick was faster than Radar Views but it also had lowest accuracy, whereas Superflick was nearly as fast and as accurate as Radar Views.

Design Recommendations and Open Questions

In this section a number of guidelines that address single user pointing are formulated. Based on the analyzed comparisons we can conclude that:

- *Local pointing*. For local pointing or when all parts of the table are within hand reach, direct techniques such as touch input [38] are preferable. To avoid the occlusion problem and for more precise positioning, touch input can be combined with zooming techniques, such as Dual finger stretch [7], or techniques that show a copy of the occluded screen area in a non-occluded location, such as Shift [23].

- *Reaching distant areas*. For reaching distant areas, techniques that bring objects closer to the user, e.g. Radar View (additional visualizations such as a pantograph line that connects input with a cursor can be used to improve the awareness) [36] and Drag-and-Pop [35] are preferable.
- *Quick transfer*. For the quick transfer of objects, throwing techniques such as Flick [19, 31, 45, 46] should be employed.

These guidelines however cannot be directly applied to a multi-user setting. In multi-user settings, speed and accuracy are less relevant performance measures than collaboration and coordination. For example, the high performance Radar View technique might not be the best choice for a collaborative setting. For more details on interaction techniques that foster coordination readers should refer to Chapter 13.

In a similar manner to selection, there are still several questions that remain unanswered:

- *Occlusion and mode switch* issues have a similar impact on pointing.
- *Multi-display*. Tabletop systems are often used in combination with other devices. In such a multi-device, multi display situation pointing becomes a more challenging task. Many techniques described here will not be able to span more than one screen. The scalability of these techniques therefore will need to be investigated further. A discussion on multi-display pointing can be found in Nacenta et al. [30].

Rotation

When people collaborate in face-to-face settings, they often share tools, artefacts and documents. The ability to reposition (translate) and reorient (rotate) content and tools is a vital part of collaboration in the tabletop setting. Tabletop systems like Entertaible © offer the potential benefit of bringing together a traditional face-to-face setting with the advantages of an electronic information processing unit. To make interaction with digital tables as flexible, intuitive and effortless as real tables we need to design techniques that allow the user to interact with a combination of digital and physical content using familiar everyday gestures. Reorienting content is one such interaction that is important for digital tables, but has not been critical for traditional desktop interaction.

In a multi-user environment, the orientation of artifacts on the table becomes even more important, as it functions as a means for communication between users. Reorientation enhances collaborative actions as it helps users to show information to other users who are seated across the table and do not share the same perspective. Further, users can orient objects on the table in such a way that they add information to their story or show whether or not the object is personal (directed towards the user) or shared (oriented towards the other users).

There are a number of challenges in designing systems and interaction techniques for tabletops; one of these is that, unlike vertical displays, tabletops do not have a predefined orientation. Since the users' perspectives change as they move around the table, designers cannot make assumptions about the preferred orientation of artefacts – there is no clear up or down, left or right. Therefore, tabletop applications must allow users to easily move and reorient objects: towards themselves for reading and manipulation, and towards others around the table during group activities.

In many tabletop and non-tabletop systems, these actions have traditionally been considered as two distinct operations (examples to follow in the next section). If one wants to move and rotate an object, one does so sequentially. On *desktop* systems, input is usually restricted to a 2DOF device like the mouse, and so rotation and translation are often divided into two separate gestures or commands. Desktop applications like Microsoft PowerPoint support planar rotation of objects by allowing the user to activate a planar rotation widget and providing rotation handles at the corner of the to-be-rotated object (Fig. 10.6). Users can rotate the object by selecting the rotation handle and moving the mouse in a clockwise or anti-clockwise direction. The object then rotates around an axis located at its centre. The translation actions are usually support via dragging.

Reorientation is an extremely common action in traditional tabletop collaborations. Kruger et al. [49] conducted an observational study of collaborative activity on a traditional table to show that the strategy of automatically reorienting objects to a person's view is overly simplistic. Their studies suggest that reorientation proves critical in how individuals *comprehend information*, how collaborators *coordinate their actions*, and how they *mediate communication*. For more details on coordination readers should refer to Chapter 13 and on collaboration to Chapter 17.

Kinaesthetic studies have demonstrated that rotating and translating are inseparable actions in the physical world [50] and that integrated interaction techniques are more appropriate mechanisms for integrated actions [51]. These studies suggest advantages for interaction methods that integrate rotation and translation to form a class of interaction techniques called *reorientation*.

Here we review various interaction techniques that have been investigated in the literature for *reorienting* content and tools on digital tables.

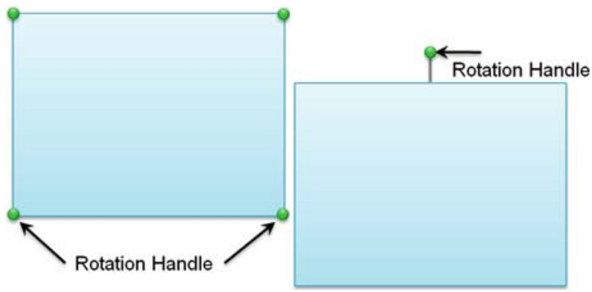

Fig. 10.6 Planar object rotation handles in standard desktop packages like Microsoft PowerPoint

Rotation Techniques

In general, there are three main ways in which orientation of objects can be supported by a tabletop system: (1) rotating the entire workspace (central part of the workspace can sometimes be fixed), (2) automatically reorienting artefacts on the workspace, or (3) allowing users to change the orientation of individual artefacts. An alternative approach is to reduce the necessity to reorient: in some cases, icons, controls and tools could also be designed in such a way that they do not have a clear orientation and can be viewed from different sides (for example, using picture icons as opposed to text labels).

There are several techniques that allow for each of these three mentioned ways of orientating objects on a tabletop [49, 52, 53]:

- *Fixed orientation*;
- *Full reorientation:* the entire physical tabletop could be either manually or electronically rotated;
- *Person-based automatic reorientation:* information is oriented automatically by the tabletop system;
- *Environment-based automatic reorientation:* items are oriented towards the person closest to it, often to the edges of a table;
- *Manual orientation:* the user to orientates the objects by themselves;
- *Multiple copies of information:* copies of information are created that can be rotated separately, so users can have their own perspective on them.

Some examples of rotation techniques are:

Corner-to-rotate. A common implementation of rotation in early tabletop systems is a direct adaptation of the desktop metaphor for 2D rotation and is used to support rotation in several tabletop systems such as DiamondSpin [54], i-LAND [55] and ConnecTables [56]. The rotation is performed by touching one of the object corners, and then turning it around an axis located at the centre of the object.

Rotate 'N Translate (RnT). RnT [57] is a tabletop interaction technique that combines rotation and translation into a single gesture. It is primarily intended for pen, stylus or finger use and uses two degrees of freedom (that correspond to the translational freedom for the input device). In RnT, the virtual object is partitioned into two regions by a circle around the object (Fig. 10.7, left). By clicking within the circular region in the centre of the artefact, the user can drag the artefact around the workspace (translate only). By clicking outside of the circle, the user can rotate and translate the artefact using a single gesture. Translation and rotation begin simultaneously, with artefact translation following the movement of the pointer.

Turn and Translate (TnT). TnT [53] is based on physical surrogates for rotation and translation of digital objects on the table, using three degrees of freedom (two

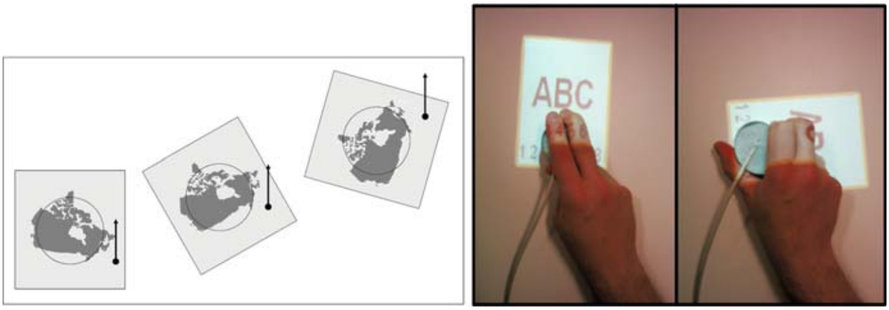

Fig. 10.7 (*Left*) RnT. A control point is selected in the lower right, and the artefact rotates counter clockwise while it is translated upward [53]. (*Right*) TnT. *Left*: sensor positioned on object. *Right*: object rotated by twisting block [53]

translational and one rotational). The user places a physical prop over the virtual object and manipulates the virtual object using the physical prop. The technique has been used in many tangible tabletop systems such as BUILD-IT [11], metaDESK [5] and Visual Interaction Platform [9], see Fig. 10.7, right.

Cutouts [58]. This technique allows users to interactively create multiple views of the tabletop workspace and move and re-orient these views.

Two finger rotation [19, 22]. This method, combined with simultaneous positioning and sometimes scaling, is the most widely used in multi-touch systems. Some systems [22] allow the user to perform reorientation and scaling using more than two fingers, in this case the system automatically uses a least-squares method to determine the motion (consisting of moving, rotating and scaling) that best satisfies the constraints provided by the fingers.

Comparison of Rotation Techniques

In an experimental study, Kruger et al. [57] compared corner-to-rotate with the RnT technique in a series of tasks that involved *precise targeting* (precise rotation and translation), *document passing* (a less-precise rotation and translation task that attempted to mirror a real-world collaborative activity – the passing of document) and *collaborative document passing* (three participants completing a word puzzle by passing and decoding clues to form a completed sentence). The authors found that RnT was faster than corner-to-rotate, and had fewer touches (but by design RnT requires one touch less than corner-to-rotate). There is no significant difference with corner-to-rotate in terms of error rate and user preference.

In a further study, Liu et al. [53] compared RnT with variations of TnT. The tasks were similar to the first two tasks of the study by Kruger et al. [57]. They found that with TnT users were almost twice as fast and twice as accurate when rotating an object, with participants indicating a greater preference for this technique.

Design Recommendations and Open Questions

The literature provides a number of formulated guidelines for supporting rotation:

- Krueger et al. [52] concluded that free rotations must be supported and the lightweight, orientation of user-positioned items must be maintained.
- Based on evaluations carried out thus far, TnT appears to be the simplest solution available that exploits tangible interaction and requires very little instruction to use.
- There has been no comparison of two-finger rotation with either the single-touch RnT or tangible rotation techniques such as TnT. But on a multi-touch table there is a strong expectation that RnT would still be a good rotation method to include alongside other techniques.
- The potential problem with using tangible objects is that reorientation could end up being an "Always ON or Always OFF" feature. At the moment there seems to be no down-side to this. But it is possible that future applications might need to explore simple ways in which the users can dynamically switch between reorientation and translation only. In this case, TnT may turn out to be no better than the corner-to-rotate technique.

There has been limited investigation into multi-touch solutions for re-orientation. It is also worthwhile to look carefully at how we can use a simple metaphor that translates across multiple input devices, such as the finger, stylus, and tangible objects. Although there have been studies on the importance of reorientation in collaboration and comprehension, most of these studies have been carried out on real tables and there have been no investigation into the effect of digital rotation techniques on collaboration and comprehension.

Scrolling

Scrolling is an important interaction technique that facilitates users' daily tasks with information spaces that are too large to be shown in their entirety, at a single point in time. Scrolling actions shift the viewport to regions that are of interest and that currently reside off-screen. There are a number of factors that influence scrolling performance, namely [6]:

- *Mapping function*. Scrolling mechanisms are driven by a mapping function that performs a translation of the user's manipulation of the input device to the scrolling operation. Various types of mapping functions exist, however most of the current systems can be classified as either a zero-order position mapping or a first-order rate mapping. In a zero-order position mapping system, the relative displacement of the cursor produces a proportional scrolling distance. This type of control is embedded within the classical scrollbar, with users controlling the viewport displacement distance by adjusting the scrollbar thumb. With a first-order rate mapping system, such as in rate-based scrolling, the mapping function translates the displacement of the cursor (or some other value such as force on an isometric input device) to scrolling speed.

- *Input device*. Several multi-stream input devices were designed to support a range of interactive tasks, including scrolling, such as a wheel or an isometric joystick [59] on a mouse.
- *Target distance*. Numerous studies have shown that scrolling mechanisms are affected by various document sizes and target distances [60–62].
- *Visual feedback*. With scrolling tasks, visual feedback typically consists of a smooth or a blurred transition from one part of the information space to another. Researchers have developed a number of alternative visualization techniques to reduce the effect of blurring that occurs at high scroll rates (e.g. [63–65]). Kaptelinin et al. [64] found that transient visual cues, an aid that differentiates the current text and the text that will replace it, can improve reading performance with text-based documents.
- *Target type*. The perceptual characteristics of the target can influence scrolling performance. Particularly at high scrolling rates, targets that are considered as being "more" preattentive facilitate more rapid searches.
- *The user's familiarity with the document*. The level of familiarity a user has with an information space has a direct effect on scrolling performance [60, 62] – users take less time to find an object if its position is known in advance.

There is large number of scrolling techniques addressing one or more of these factors. To provide focus, we limit our discussion to conventional scrolling techniques used on desktop systems, and scrolling techniques designed specifically for tabletop systems.

Scrolling Techniques

Scrolling techniques can be classified into two categories: device independent and pen/touch-based techniques.

Device independent: These scrolling techniques are controlled using a variety of input devices. The *scrollbar* is the most commonly employed interface widget for navigating large information spaces, requiring the user to position a scrollbar "thumb"; *rate-based scrolling* uses a first-order rate mapping, the mapping function translates the displacement of the cursor (or some other value such as force on an isometric input device) to scrolling speed; *panning* (e.g. using Adobe Reader's *hand tool)* facilitates 2D scrolling dragging the face of the content to the required position; *Space-Filling Thumbnails (SFT)* [61] allows users to switch between a normal reading view and a thumbnail view in which all document pages are scaled and tiled to fit the window, providing an overview of all of the document's pages and ultimately eliminates the need for a scrollbar.

Pen/touch-based scrolling: The *scroll ring* (e.g. Apple iPod^{TM}) is designed as a doughnut-shaped touchpad that scrolls content in response to a user's circular strokes; the *virtual scroll ring* [66] is a software implementation of the scroll ring, negating the need for additional hardware; the *radial scroll tool* [67] uses a stirrer (a device that converts circular motions of a pen into a series of values that can be used to control the rotation of an object, as studied by Evans et al. [68]) as a tool to determine the direction in which the pen is spinning, feeding the output into scroll direction and velocity; *curve dial* [69] is an improved version of the radial scroll tool, it uses the curvature of the mouse drag (or drag of another pointing device such as pen or finger) to determine the direction of scrolling; *crossbar* [28] allows users to navigate by crossing the pen over a crossbar; *gesture scrolling/panning* [22] is based on a "hand" tool, but is enriched by multi-touch capabilities: in a similar manner to the hand tool, users can start scrolling by sliding a finger along the surface, while the speed of scrolling is controlled by the number of fingers in contact with surface, i.e. scrolling speed increases as the number of fingers increases; *Multi-Flick-Standard* (MFS) [6] maps the pen flick speed to the document scrolling speed, similar to setting a flywheel in motion; *Compound Multi-Flick* (CMF) [6] is a compound technique that combines flicking with a displacement-based control; *Multi-Flick-Friction* (MFF) [6] is similar to MFS but includes an additional friction "factor" that gracefully reduces the document scrolling speed after some time interval.

Comparison of Scrolling Techniques

Smith and Schraefel [67] compared the Radial Scroll Tool with standard scrolling techniques available in touch-based systems. The independent variables were interface type (tablet and large screen touch display), scroll distance (short: targets 5 pages/images apart and long: targets 20 pages/images apart), technique (radial scroll and traditional scroll, namely the scrollbar in combination with software keyboard's arrow and page keys), and task type (find a picture, find a heading). The study showed that for both the tablet and the wall mounted display, with both image and text selection, radial scroll worked better for navigating short distances and traditional scroll worked better for long distances.

Aliakseyeu et al. [6] compared three multi-flick scrolling technique modifications with the standard scrollbar on three pen based devices – a table, a Tablet PC and a PDA. Two experiments were carried out: one that required users to scroll a list of items (names of cities), and another that required scrolling in a standard text document. They concluded that Multi-flick scrolling is an effective alternative to conventional scrolling techniques. They found that Compound-Multi Flick is on average faster than the two other flick modifications and is as effective as using the scrollbar and is most preferred by users.

Design Recommendations and Open Questions

Scrolling short documents (less than five pages):

- In situations where the scrolling distances are short and the user requires good control of the scrolling speed (for example, when reading), techniques similar to Curve Dial [69] are the most appropriate.
- In situations where the document is unknown and the user wishes to explore the information space (for example, browsing a web page) multi-flick [6] based techniques are preferable.

Scrolling long documents:

- When scrolling through a long and unknown document, multi-flick [6] based techniques are a good option. These techniques allow scrolling to start with a single stroke. The user can then remove their hands from the table, avoiding the occlusion problem.
- When scrolling through known documents, the Crossbar [28] is an appropriate choice. It provides the user with information regarding their position within the document, it allows them to quickly jump into a desired location and it is based on a common and well know widget (the scrollbar). However, this technique is only appropriate for 1D scrolling (the introduction of the second bar will break the scrolling into two separate actions: vertical and horizontal scrolling). Alternatives such as multi-flick may be better suited when 2D scrolling is required.
- A standard scrollbar can be used for scrolling though long, but familiar documents. In a similar manner to the crossbar, the scrollbar can only facilitate 1D scrolling.

To provide a complete solution, all three of the techniques mentioned above can be implemented simultaneously. Users can easily discriminate between curve dial and flick and the crossbar provides a separate interface element. This leaves the user to choose the most appropriate or their preferred option depending on their task at hand.

There are still a number of questions that remain unanswered:

- *Mode switch*. Some interaction techniques may require a mode switch to discriminate them from application commands. For example, in a drawing program, a flicking gesture for drawing a line needs to be discriminated from a flicking gesture to pan the canvas. Curve Dial and Multi-Flick may require this type of mode switch.
- *Finger friction*. Curve Dial requires a constant circular movement of the finger, which is potentially a tiring action and should probably only be used for scrolling short distances. Alternatively, tangible objects can be used instead of the finger to reduce the friction between the finger and the table.

- *Screen real-estate and document position awareness*. The scrollbar, while taking up some valuable screen real-estate, provides the user with useful information regarding their location within the document and the current portion of the document on-screen. Even when employing gesture-based scrolling techniques, users still require some awareness of their position. A passive scrollbar (i.e. one that cannot be interacted with) can be employed in this situation; however, it remains to be seen if this is the best solution.
- *Tangible scrolling*. Instead of a finger, an active or passive tangible object can be used for scrolling. For example, a tangible object with a wheel similar to the mouse wheel can be used for this purpose – the location of the object would define the document to scroll and turning of the wheel would perform the scrolling itself (the wheel turn can either be detected by the table – passive tangible, or it can be done electronically within the object and transmitted wirelessly to the table – active tangible).
- *Visual enhancement techniques* (SFT, SDAZ, etc.). A number of techniques have been proposed to allow a higher scrolling speed by using different visualization approaches [28, 63–65]. So far these techniques have been implemented and studied solely on desktop systems, their usability on tabletop systems is unclear and requires further investigation.

Discussion

In presenting the interaction techniques we also identify design factors (or issues) that affect each atomic action. For example, selection is affected by *reaching distance or range, number of states of selection model, occlusions* and *parallax*.

Although each section also identifies how interaction techniques address some of these issues, it is not always clear how well a particular technique deals with each design factor or if any of the factors have a negative impact on the technique.

One further limitation of the techniques documented in this chapter is the lack of evaluation in terms of multi-user coordination, collaboration and comprehension. The most commonly used performance measures are time, error-rate and user preference. However, in a multi-user setting other metrics that relate to group dynamics are just as important. More research is needed to identify design principles and guidelines that take into consideration the above factors (interested readers should refer to Chapter 13 for details).

This chapter focused only on four types of atomic actions – selection, pointing, rotation and scrolling. Actions such as menu navigation, 2D map navigation, and text-entry are equally important although due to space limitations these are not explored within this chapter.

Conclusion

In this chapter we have provided a structured review of the research efforts into supporting atomic user actions. Specifically, we looked at selecting, pointing, rotating and scrolling actions. Our review considered input methods supported by different tabletop systems and also identified challenges faced when designing interaction techniques that support these atomic actions. Despite the large number of interaction techniques that offer different support for atomic actions, there are no clearly preferred techniques or techniques that address all design challenges. Based on this review we identify several open research questions that we hope will stimulate further research in this area.

References

1. Wellner P (1993) Interacting with paper on the DigitalDesk. Communications of the ACM 36(7):87–96, doi: 10.1145/159544.159630
2. Arai T, Machii K, Kuzunuki S, Shojima H (1995) InteractiveDESK: A computer-augmented desk which responds to operations on real objects. In: CHI'95: Proceedings of the conference companion on human factors in computing systems, ACM Press, New York, pp 141–142, doi: 10.1145/223355.223470
3. Krüger W, Bohn CA, Fröhlich B, Schüth H, Strauss W, Wesche G (1995) The responsive workbench: A virtual work environment. Computer 28(7):42–48, doi:10.1109/2.391040
4. Fitzmaurice GW, Ishii H, Buxton WAS (1995) Bricks: Laying the foundations for graspable user interfaces. In: CHI'95: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., New York, pp 442–449, doi: 10.1145/223904.223964
5. Ullmer B, Ishii H (1997) The metaDESK: Models and prototypes for tangible user interfaces. In: UIST'97: Proceedings of the 10th annual ACM symposium on User interface software and technology, ACM Press, New York, pp 223–232, doi: 10.1145/263407.263551
6. Aliakseyeu D, Irani P, Lucero A, Subramanian S (2008) Multi-flick: An evaluation of flick-based scrolling techniques for pen interfaces. In: CHI'08: Proceeding of the twenty-sixth annual SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 1689–1698, doi: 10.1145/1357054.1357319
7. Benko H, Wilson A, Baudisch P (2006) Precise selection techniques for multi-touch screens. In: CHI'06: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 1263–1272, doi: 10.1145/1124772.1124963
8. Dietz P, Leigh D (2001) DiamondTouch: A multi-user touch technology. In: UIST'01: Proceedings of the 14th annual ACM symposium on User interface software and technology, ACM Press, New York, pp 219–226, doi: 10.1145/502348.502389
9. Aliakseyeu D, Martens J-B, Subramanian S, Vroubel M, Wesselink W (2001) Visual interaction platform. In: INTERACT'01: Proceedings of Interact 2001, IOS Press, Amsterdam, pp 232–239
10. Brandl P, Forlines C, Wigdor D, Haller M, Shen C (2008) Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In: AVI'08: Proceedings of the working conference on advanced visual interfaces, ACM Press, New York, pp 154–161, doi: 10.1145/1385569.1385595
11. Rauterberg M, Fjeld M, Krueger H, Bichsel M, Leonhardt U, Meier M (1997) BUILD-IT: A computer vision-based interaction technique for a planning tool. In: Proceedings of the HCI on people and computers XII, Springer-Verlag, London, pp 303–314
12. Balakrishnan R (2004) "Beating" Fitts' law: Virtual enhancements for pointing facilitation. International Journal of Human-Computer Studies 61(6):857–874, doi: 10.1016/j.ijhcs.2004.09.002
13. MacKenzie IS (1992) Fitts' law as a research and design tool in human-computer interaction. Human-Computer Interaction 7(1):91–139, doi: 10.1207/s15327051hci0701_3
14. Bohan M, Chaparro A (1998) To click or not to click: A comparison of two target-selection methods for HCI. In: CHI'98: Proceedings of the CHI 98 conference summary on human factors in computing systems, ACM Press, New York, pp 219–220, doi: 10.1145/286498.286707

15. Hourcade JP, Berkel TR (2006) Tap or touch? Pen-based selection accuracy for the young and old. In: CHI'06: Extended abstracts of the CHI' 06 conference on human factors in computing systems, ACM Press, New York, pp 881–886, doi: 10.1145/1125451.1125623
16. Buxton WAS (1990) A three-state model of graphical input. In: INTERACT'90: Proceedings of the interact '90, North-Holland Publishing Co., Amsterdam, pp 449–456
17. Potter RL, Weldon LJ, Shneiderman B (1988) Improving the accuracy of touch screens: An experimental evaluation of three strategies. In: CHI'88: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 27–32, doi: 10.1145/57167.57171
18. MacKenzie IS, Oniszczak A (1998) A comparison of three selection techniques for touchpads. In: CHI'98: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., New York, pp 336–343, doi: 10.1145/274644.274691
19. Wu M, Balakrishnan R (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: UIST'03: Proceedings of the 16th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 193–202, doi: 10.1145/964696.964718
20. Masoodian M, McKoy S, Rogers B (2007) Hands-on sharing: Collaborative document manipulation on a tabletop display using bare hands. In: CHINZ '07: Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on computer-human interaction: Design centered HCI, ACM Press, New York, pp 25–31, doi: 10.1145/1278960.1278964
21. Esenther A, Ryall K (2006) Fluid DTMouse: Better mouse support for touch-based interactions. In: AVI'06: Proceedings of the working conference on advanced visual interfaces, ACM Press, New York, pp 112–115, doi: 10.1145/1133265.1133289
22. Rekimoto J (2002) SmartSkin: An infrastructure for freehand manipulation on interactive surfaces. In: CHI'02: Proceedings of the SIGCHI conference on human factors in computing systems: Changing our world, changing ourselves, ACM Press, New York, pp 113–120, doi: 10.1145/503376.503397
23. Vogel D, Baudisch P (2007) Shift: A technique for operating pen-based interfaces using touch. In: CHI'07: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 657–666, doi: 10.1145/1240624.1240727
24. Wigdor D, Penn G, Ryall K, Esenther A, Shen C (2007) Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In: Tabeltop'08: Proceedings of the second annual IEEE international workshop on horizontal interactive human-computer systems, pp 60–67, doi: 10.1109/TABLETOP.2007.33
25. Wigdor D, Shen C, Forlines C, Balakrishnan R (2006) Table-centric interactive spaces for real-time collaboration. In: AVI'06: Proceedings of the working conference on advanced visual interfaces, ACM Press, New York, pp 103–107, doi: 10.1145/1133265.1133286
26. Olwal A, Feiner S, Heyman S (2008) Rubbing and tapping for precise and rapid selection on touch-screen displays. In: CHI'08: Proceedings of the twenty-sixth annual SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 295–304, doi: 10.1145/1357054.1357105
27. Ren X, Moriya S (2000) Improving selection performance on pen-based systems: A study of pen-based interaction for selection tasks. ACM Transactions on Computer-Human Interaction 7 (3): 384–416, doi: 10.1145/355324.355328
28. Apitz G, Guimbretière F (2004) CrossY: A crossing-based drawing application. In: UIST'04: Proceedings of the 17th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 3–12, doi: 10.1145/1029632.1029635
29. Aliakseyeu D, Martens JB, Rauterberg M (2006). A computer support tool for the early stages of architectural design. Interacting with Computers 18(4):528–555, doi: 10.1016/j.intcom.2005.11.010
30. Nacenta M, Gutwin C, Aliakseyeu D, Subramanian S (2009) There and back again: Cross-display object movement in multi-display environments. Human-Computer Interaction 24(1&2):170–229, doi: 10.1080/07370020902819882

31. Geißler J (1998) Shuffle, throw or take it! Working efficiently with an interactive wall. In: CHI'98: Proceedings of the CHI 98 conference summary on human factors in computing systems, ACM Press, New York, pp 265–266, doi: 10.1145/286498.286745
32. Baudisch P, Cutrell E, Robbins D, Czerwinski M, Tandler P, Bederson B, Zierlinger A (2003) Drag-and-pop and Drag-and-pick: Techniques for accessing remote screen content on touch and pen-operated systems. In: INTERACT'03: Proceedings of the Interact'03, Springer, pp 57–64
33. Albinsson P, Zhai S (2003) High precision touch screen interaction. In: CHI'03: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 105–112, doi: 10.1145/642611.642631
34. Nacenta M, Sallam S, Champoux B, Subramanian S and Gutwin C (2006) Perspective cursor: Perspective-based interaction for multi-display environments. In: CHI'06: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 289–298, doi: 10.1145/1124772.1124817
35. Collomb M, Hascoët M, Baudisch P, Lee B (2005) Improving drag-and-drop on wall-size displays. In: GI'05: Proceedings of the graphics interface 2005, Canadian Human-Computer Communications Society, pp 25–32
36. Nacenta MA, Aliakseyeu D, Subramanian S, Gutwin C (2005) A comparison of techniques for multi-display reaching. In: CHI'05: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 371–380, doi: 10.1145/1054972.1055024
37. Nacenta MA, Pinelle D, Stuckel D, Gutwin C (2007) The effects of interaction technique on coordination in tabletop groupware. In: GI'07: Proceedings of the graphics interface 2007, ACM Press, New York, pp 191–198, doi: 10.1145/1268517.1268550
38. Ha V, Inkpen K, Mandryk R, Whalen T (2006) Direct intentions: The effects of input devices on collaboration around tabletop display. In: Tabletop'06: Proceedings of the first IEEE international workshop on horizontal interactive human-computer systems, IEEE, pp 177–184, doi: 10.1109/TABLETOP.2006.10
39. Rekimoto J (1997) Pick-and-drop: A direct manipulation technique for multiple computer environments. In: UIST'97: Proceedings of the 10th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 31–39, doi: 10.1145/263407.263505
40. Matejka J, Grossman T, Lo J, Fitzmaurice GW (2009) The design and evaluation of multi-finger mouse emulation techniques. In: CHI'07: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 1073–1082, doi: 10.1145/1518701.1518865
41. Grossman T, Balakrishnan R (2005) The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In: CHI'05: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 281–290, doi: 10.1145/1054972.1055012
42. Worden A, Walker N, Bharat K, Hudson S (1997) Making computers easier for older adults to use: Area cursors and sticky icons. In: CHI'97: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 266–271, doi: 10.1145/258549.258724
43. Chapuis O, Labrune J-B, Pietriga E (2009) DynaSpot: Speed-dependent area cursor. In: CHI'09: Proceedings of the 27th international conference on human factors in computing systems, ACM Press, New York, pp 1391–1400, doi: 10.1145/1518701.1518911
44. Aliakseyeu D, Nacenta MA, Subramanian S, Gutwin C (2006) Bubble radar: Efficient pen-based interaction. In: AVI'06: Proceedings of the working conference on advanced visual interfaces, ACM Press, New York, pp 19–26, doi: 10.1145/1133265.1133271
45. Hascoët M (2003) Throwing models for large displays. In: Proceedings of the British HCI 2003, British HCI Group, pp 73–77
46. Moyle M, Cockburn A (2002) Analyzing mouse and pen flick gestures. In: Proceedings of the SIGCHI-NZ, pp 266–267

47. Reetz A, Gutwin C, Stach T, Nacenta MA, Subramanian S (2006) Superflick: A natural and efficient technique for long-distance object placement on digital tables. In: GI'06: Proceedings of the graphics interface 2006, Canadian Information Processing Society, pp 163–170
48. Forlines C, Vogel D, Balakrishnan R (2006) HybridPointing: Fluid switching between absolute and relative pointing with a direct input device. In: UIST'06: Proceedings of the 19th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 211–220, doi: 10.1145/1166253.1166286
49. Kruger R, Carpendale S, Scott SD, Greenberg S (2003) How people use orientation on tables: Comprehension, coordination and communication. In: CSCW'03: Proceedings of the 2003 international ACM SIGGROUP conference on supporting group work, ACM Press, New York, pp 369–378, doi: 10.1145/958160.958219
50. Yanqing W, Christine LM, Valerie AS, Kellogg SB (1998) The structure of object transportation and orientation in human-computer interaction. In: CHI'98: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., New York, pp 312–319, doi: 10.1145/274644.274688
51. Jacob R, Sibert LE, McFarlane DC, Mullen J (1994) Integrality and separability of input devices. ACM Transactions on Computer-Human Interaction 1(1):3–26, doi: 10.1145/174630.174631
52. Kruger R, Carpendale S (2003) Exploring orientation on a table display. Department of Computer Science, University of Calgary
53. Liu J, Pinelle D, Sallam S, Subramanian S, Gutwin C (2006) TNT: Improved rotation and translation on digital tables. In: GI'06: Proceedings of the graphics interface, Canadian Information Processing Society, pp 25–32
54. Shen C, Frédéric DV, Forlines C, Ringel M (2004) DiamondSpin: An extensible toolkit for around-the-table interaction. In: CHI'04: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 167–174, doi: 10.1145/985692.985714
55. Streitz NA, Geißler J, Holmer T, Konomi S, Müller-Tomfelde C, Reischl W, Rexroth P, Seitz P, Steinmetz R (1999) i-LAND: An interactive landscape for creativity and innovation. In: CHI'99: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 120–127, doi: 10.1145/302979.303010
56. Tandler P, Prante T, Müller-Tomfelde C, Streitz N, Steinmetz R (2001) Connectables: Dynamic coupling of displays for the flexible creation of shared workspaces. In: UIST'01: Proceedings of the 14th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 11–20, doi: 10.1145/502348.502351
57. Kruger R, Carpendale S, Scott S, Tang A (2005) Fluid integration of rotation and translation. In: CHI'05: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 601–610, doi: 10.1145/1054972.1055055
58. Pinelle D, Dyck J, Gutwin C, Stach T (2006) Cutouts: Multiple views for tabletop groupware. HCI-TR-06-04, Computer Science Department, University of Saskatchewan
59. Zhai S, Smith BA, Selker T (1997) Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In: Proceedings of the IFIP TC13 international conference on human-computer interaction, IFIP conference proceedings, vol 96. Chapman & Hall Ltd., London, pp 286–293
60. Andersen TH (2005) A simple movement time model for scrolling. In: CHI'05: Extended abstracts of the CHI '05 conference on human factors in computing systems, ACM Press, New York, pp 1180–1183, doi: 10.1145/1056808.1056871
61. Cockburn A, Gutwin C, Alexander J (2006) Faster document navigation with space-filling thumbnails. In: CHI'06: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 1–10, doi: 10.1145/1124772.1124774
62. Hinckley K, Cutrell E, Bathiche S, Muss T (2002) Quantitative analysis of scrolling techniques. In: CHI'02: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 65–72, doi: 10.1145/503376.503389

63. Cockburn A, Savage J, Wallace A (2005) Tuning and testing scrolling interfaces that automatically zoom. In: CHI'05: Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press, New York, pp 71–80, doi: 10.1145/1054972.1054983
64. Kaptelinin V, Mäntylä T, Åström J (2002) Transient visual cues for scrolling: An empirical study. In: CHI'02: Extended abstracts of the CHI '02 conference on human factors in computing systems, ACM Press, New York, pp 620–621, doi: 10.1145/506443.506513
65. Klein C, Bederson BB (2005) Benefits of animated scrolling. In: CHI'05: Extended abstracts of the CHI '05 conference on human factors in computing systems, ACM Press, New York, pp 1965–1968, doi: 10.1145/1056808.1057068
66. Moscovich T, Hughes JF (2004) Navigating documents with the virtual scroll ring. In: UIST'04: Proceedings of the 17th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 57–60, doi: 10.1145/1029632.1029642
67. Smith GM, Schraefel MC (2004) The radial scroll tool: Scrolling support for stylus- or touch-based document navigation. In: UIST'04: Proceedings of the 17th annual ACM symposium on user interface software and technology, ACM Press, New York, pp 53–56, doi: 10.1145/1029632.1029641
68. Evans KB, Tanner PP, Wein M (1981) Tablet-based valuators that provide one, two, or three degrees of freedom. ACM SIGGRAPH Computer Graphics 15(3):91–97, doi: 10.1145/965161.806794
69. Smith G, Schraefel MC, Baudisch P (2005) Curve dial: Eyes-free parameter entry for GUIs. In: CHI'05: Extended abstracts of the CHI '05 conference on human factors in computing systems, ACM Press, New York, pp 1146–1147, doi: 10.1145/1056808.1056855