Attribute Gates

Ahmed N. Sulaiman
Culture Lab
School of Computing Science
Newcastle University
Newcastle upon Tyne
NE1 7RU, UK
ahmed.sulaiman@ncl.ac.uk
Telephone: +971-50-5095301

Patrick Olivier
Culture Lab
School of Computing Science
Newcastle University
Newcastle upon Tyne
NE1 7RU, UK
p.l.olivier@ncl.ac.uk
Telephone: +44-191-246-4630

ABSTRACT

Attribute gates are a new user interface element designed to address the problem of concurrently setting attributes and moving objects between territories on a digital tabletop. Motivated by the notion of task levels in activity theory, and crossing interfaces, attribute gates allow users to operationalize multiple subtasks in one smooth movement. We present two configurations of attribute gates; (1) *grid gates* which spatially distribute attribute values in a regular grid, and require users to draw trajectories through the attributes; (2) *polar gates* which distribute attribute values on segments of concentric rings, and require users to align segments when setting attribute combinations. The layout of both configurations was optimised based on targeting and steering laws derived from Fitts' Law. A study compared the use of attribute gates with traditional contextual menus. Users of attribute gates demonstrated both increased performance and higher mutual awareness.

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

General terms: Design

Keywords: Digital tabletops, crossing interfaces, pen-based input, large interactive displays, tabletop territories, user interface components.

INTRODUCTION

When people collaborate around a digital tabletop, the surface is typically divided into personal, public, and storage territories [19]. Different values for size, orientation, and access rights of documents and other objects are used for different territories. For example, users working on a document in their personal space might typically want: (1) full access rights; (2) the document to be oriented towards them; and (3) the document to be sized appropriately for the task at hand (i.e. large for editing, smaller for scanning). On the other hand, users moving an object to the public space (for other users to store) might typically want to set the document to be: (1) read-only access; (2) oriented away from them (i.e. presented to other users); and (3) iconified (i.e. to save space). The large number of attribute value combinations needed, reflects the varied nature of the objects used and work performed at tables. As Kruger et al. [14] observed, users of digital tabletops frequently override default territory-based attribute values.

A number of recommendations have been made regarding these attributes. For orientation, these include automatic and manual rotation, or the use of rotate-and-translate techniques [14, 15, 19 and 21]. Proposal for size include manual resizing [21] and increasing or decreasing size with respect to the center of the workspace (i.e. fish-eye techniques). As with conventional files and documents, access privileges, can be set to allow an object to be fully accessible, read-only, or simply a copy of an original [16, 18, and 20].

Digital tabletop interactions are usually pen- or finger-based and these input methods are not well suited to traditional point-and-click techniques developed for mouse input. Point-and-click interaction has a tendency to segment any operation into a series of discrete actions. This contrasts with pen input, which encourages a fluid, stroke-based style of interaction [4]. Crossing-based interfaces have been proposed as an alternative to point-and-click techniques for pen-based input [3]. For example, the *CrossY* application for tablet PCs [4], and the *Interactive Mural* for interactive whiteboards [22]. The *FlowMenu* [9] developed initially for the *Interactive Mural* has also been used in digital tabletop systems where it was found to be an improvement on traditional point-and-click based context menus [19].

Tabletops have some other unique characteristics that have implications for the design of its interface components. These include the large-horizontal space, and the multi-user environment. The design of attribute gates reflects these special characteristics.

Attribute gates (figure 1) are interface elements, designed to be used for pen- or finger-based interaction on large horizontal interactive surfaces, to set a sequence of scale,

UIST'08, October 19–22, 2008, Monterey, California, USA.

orientation and access right attributes in one fluid operation.

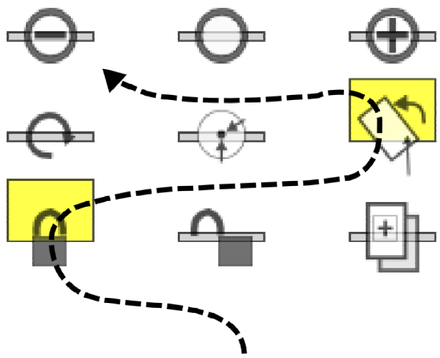

Figure 1(a): Grid gates. The dashed curve shows the user crossing two gates (read-only, and rotate and translate) and heading for the third (reduce size).

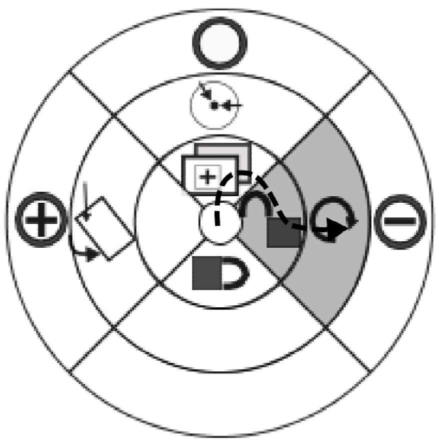

Figure 1(b): Polar gates. The dashed curve shows the user crossing to a gate in the inner ring (read/write) and rotating the ring towards a gate in the second ring (manual rotate) and moving outward to the third (reduce size).

Attribute gates are motivated by the need for a fluid way to change a number of different settings when moving objects between territories on the table. The collaborative nature of digital tabletops requires components to promote the mutual awareness of users. This is a particular concern for multi-user environments, yet is often ignored in the application of desktop interface components to tabletops. The design of attribute gates is based on both the principle of task levels from activity theory, and the notion of crossing-based interfaces. Moreover, the spatial configuration of attribute gates can be optimized through the application of targeting and steering laws derived from Fitts' law and consideration of the large horizontal surface of a digital tabletop. We have conducted a number of user studies to evaluate the two types of attribute gates and have compared this with traditional contextual menus in terms of performance, accuracy, and mutual awareness.

MOTIVATION

Kruger et al. [14] and Scott [19] suggested a number of design guidelines for the rotation of objects on digital tabletops. These emphasized the importance of supporting both lightweight free rotation techniques, and automatic rotation performed by the system. Kruger et al. [14] noted that even for systems that support automatic orientation, there are many situations in which users need a different orientation to the system's default. Scott [19] generalized this recommendation to incorporate scaling. Morris' [16] guidelines for designing digital tabletop systems included a requirement for providing a means of dynamically changing document access rights on the tabletop, and recommended making these access rights visible to increase awareness, thereby enabling users to regulate participation levels and prevent confusion.

Though it has been made clear that a tabletop interface must provide lightweight techniques for dynamically controlling orientation, scaling, and access rights, only a few specific methods have been proposed. Scott [19] presented an approach to rotation whereupon moving an object, the system provided automatic behavior that included a preview of the proposed action. The user could then invoke a context menu (Scott actually used a *FlowMenu* [9]) offering options to accept-the-default, ignore-the-default, free-rotate, and rotate left, top, bottom, or right. As well as only addressing rotation (and not other attributes such as size and access rights) this technique lacks the fluid and lightweight characteristics that attribute gates seek to realize.

A user moving a document from her personal space to the public space might reasonably wish to set the document attributes to read-only, manual rotate, and enlarge. A traditional contextual menu would incorporate three distinct commands and sub-commands for the different settings of each attribute (assuming a discrete set of values for each). Thus the user needs to select six commands and subcommands to change the attributes. This is at best off-putting, and at worst encourages the user to stick with inappropriate defaults. The provision of a means of setting the required orientation, size, and access right attributes in one fluid operation, while moving objects between tabletop territories, is the principal design goal of attribute gates. In addition, the design of attribute gates seeks to promote mutual awareness at the tabletop, in particular in relation to access right attributes, which are usually only visible when a user browses an object's properties.

Finally, in creating a user interface component specifically for digital tabletops we have sought to exploit the unique characteristics of large horizontal surfaces that use pen or finger input. With the large footprint of the digital tabletop, and the coincidence of the action and perception spaces, targeting objects and steering along deliberate paths is considerably easier than on a tablet PC [2], or mouse-based desktop or laptop computers.

THE THEORETICAL BASIS OF ATTRIBUTE GATES

There is a substantial theoretical and empirical interaction design literature that can be used to both explain the role and utility of attribute gates, and optimally configure their spatial layout and dimensions.

Activity theory and chunking

Activity theory [13] is a useful framework for analyzing physical and co-located collaborative interaction settings. Here *activity* is defined as the minimal meaningful context that is directed to an object in order to transform it into an outcome. An activity is the basic unit of analysis, driven by a motive, and is carried out by a series of actions. An *action* is a conscious act with a direct defined goal that usually consists of a number of operations. An *operation* is the subconscious act that might once have been an action done consciously, but with practice and repetition became a routine act (was turned into an operation). For example, when a person with little experience of using a QWERTY keyboard wants to type text, locating each letter is an action in itself. However, when this user gains experience, typing becomes a series of operations performed subconsciously.

Individuals new to a certain activity need to think about every step of the process. Such a process is undertaken through a series of well thought out actions, with clear specific intentions behind each. In such a case, no operations are involved and focus shifts from the high-level task to trivial low-level tasks. After practice and repetition, appropriate actions are performed subconsciously, and this transformation to operations (in the user's head) allows them to focus on higher level tasks. As more actions are turned into operations, it becomes easier to stop worrying about the details and concentrate on the desired outcome as the individual will subconsciously trigger the appropriate sequence of operations depending on the conditions at hand. Kaptelinin [11] observed that by looking at whether a subject's behavior, in a specific situation, is oriented toward a motive, a goal, or is in response to a specific condition, one can better understand and predict the subject's behavior.

In simple terms, activity theory tells us that good user interface components move actions into operations. Buxton's work on chunking [5] is strikingly similar to activity theory. In discussing the differences between the levels of detail that novices and experts attend to, he describes how for novices, finding a character on the keyboard or remembering the name of a command, requires valuable cognitive resources (which can be performed by experts automatically). Where activity theory describes the progression from novice to expert in terms of carrying out more actions as operations, Buxton's notion of chunking refers to the amount of a problem that can be performed automatically (i.e. as an operation) and he proposes gluing a number of subtasks into one task. According to Buxton, the three subtasks required to select a command from a contextual menu can be glued together if a simple modification is made where the user presses and holds the right button, moves to the desired command, then releases the mouse button. In this case the tension of pressing the mouse button is the "glue".

Crossing-based interfaces

Accot and Zhai [3] proposed *crossing* as an alternative to point-and-click interfaces, especially for pen-based interaction. Crossing allows the initiation of a command by simply crossing a specific target (represented by a spatial location) without the need to point or click on interface components. Accot and Zhai found that goal crossing could be more efficient, or at least as efficient, as pointing. They provided a number of guidelines for designing crossing-based interfaces and recommended that whenever possible, the goal to be crossed should be orthogonal to the direction of movement. With crossing, selecting a command is one task that cannot be subdivided into further subtasks, and if a number of related commands are positioned appropriately, it is possible to issue more than one command in a single fluid movement. By appropriately positioning commands, crossing-based interfaces have the potential to satisfy the design recommendations suggested by both activity theory [13] and chunking [5].

Targeting and steering

Attribute gates use the crossing principle, and setting attributes involves steering between elements and crossing others. The layout of these elements may be optimized (to increase ease of use and efficiency) by the application of targeting and steering laws derived from Fitts' law [8].

Accot and Zhai [1] extended Fitts' law through the introduction of an equation to calculate the time required for steering inside a path. Assuming a path of fixed width *W* and of length *D*, the time (*T*) required to move inside that path is:

$$T = a + b\left(\frac{D}{W}\right) \qquad (1)$$

Where *a* and *b* are empirically established constants that are characteristic of a user.

In equation (1), the ratio *D/W* is the index of difficulty. Steering time has a linear relation with the *D/W* ratio, unlike Fitts' original formulation, in which the targeting time has a logarithmic relation to *D/W*. Accot and Zhai have demonstrated how this law can be used to estimate the time required to select and navigate through commands in a multi-level menu structure. Another investigation of steering tasks was carried out by Drury [7], who studied vehicle guidance tasks in linear and circular paths, although the equation proposed by Accot and Zhai is simpler and more relevant to the task at hand. With this in mind, attribute gates are designed to both optimize the targeting and steering effort in combination with the operationalization of actions proposed by activity theory.

ATTRIBUTE GATES: INTERACTION

Attribute gates make the process of setting a sequence of attributes a fluid operation that requires no shift of focus from the main activity (moving an object between territories). Two types of attribute gate with different spatial characteristics are proposed, grid gates and polar gates**.** Gates are laid out so as to: position mutually exclusive attributes together; spatially sequence different groups; and allow users to set an attribute simply by crossing it while a document or object is being dragged towards its destination. The design of the gates makes use of two key characteristics of digital tabletops:

- The coincidence of action and perception space: provides the user with a sense of control over the operation.
- The large surface area of the table: allows the user to easily move across the desired attributes without careful maneuvering.

Furthermore, integrating attribute value assignment with movement (which is the key element) means that users are not forced to change their focus or type of action, or perform additional actions (such as right clicks).

Grid Gates

In a grid gate, each group of mutually exclusive attributes is placed in a row. Each attribute is represented by a thin rectangular area, and the rectangles are separated by empty space. The overall layout is a grid of rectangles (see figure 2).

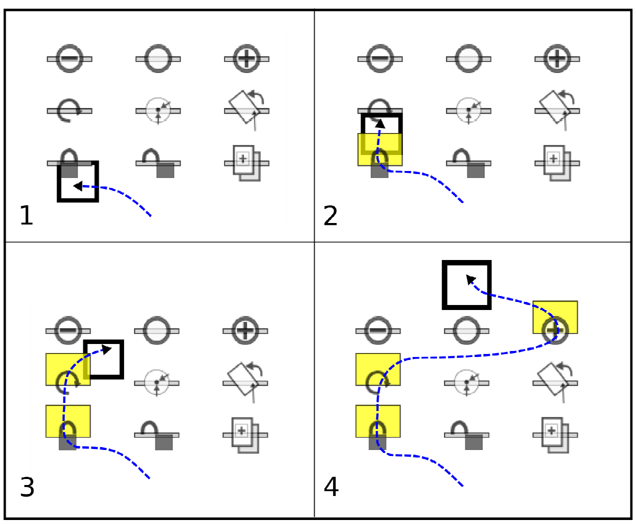

Figure 2: The steps for setting the read-only, manual-rotate, and enlarge attribute values using a grid gate.

To set a document's attributes to read-only, manual rotate, and enlarge, the user needs to press and hold on the document for half-a-second to display the grid gate (this duration was chosen on the basis of informal experimentation) (fig. 2-1). Next the user needs to pass the dragging point through (over) the required gates. In this case these are the 'read-only' (fig. 2-2), 'manual rotate' (fig. 2-3), and finally 'enlarge' gates (fig. 2-4). As soon as the user releases pen pressure, the gate disappears. If the user wants to set only one or two attributes, she can pass through the empty space of the other rows to maintain their default value. Also, if the user wants to change one of the attributes after setting it, and before the gates disappear, she can re-cross the new required setting (passing through the empty spaces for the others attributes).

Polar Gates

In the polar setting, groups of mutually exclusive attributes are arranged in concentric rings (see figure 3). An important behavior of the polar gate is that when a ring is rotated, it rotates the inner rings only, leaving the outer rings unchanged. By not resetting the orientation of the rings between activations, the polar gate effectively 'remembers' its last setting.

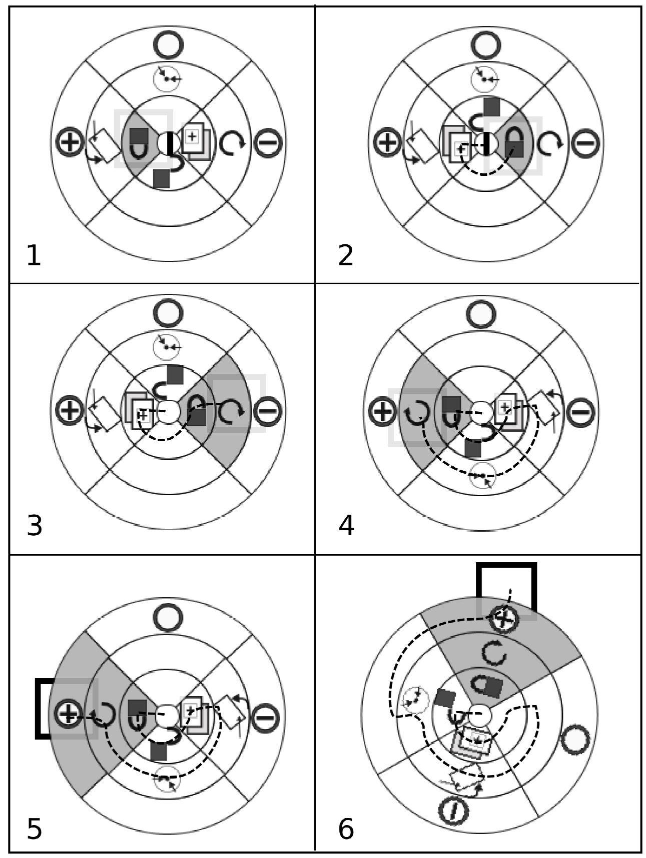

Figure 3: The steps for setting the read-only, manual-rotate, and enlarge attribute values using polar gates.

To set a document's attributes to read-only, manual rotate, and enlarge, the user needs to press and hold on the document for half-a-second. This displays the polar gate centered on the pen's tip. The user then needs to move towards the 'read-only' gate (fig. 3-1), rotate the ring towards the manual rotate gate (fig. 3-2), move over that gate (fig. 3-3) then rotate the ring towards the enlarge gate (fig. 3-4). When passing over the 'enlarge' gate (fig. 3-5) the user can rotate the ring in the direction of the public space then continue moving the document to its final destination (fig. 3-6). The gate disappears when the user releases the pen pressure.

When the polar gate next appears, it shows the most recently used setting, with the segments aligned in the direction of the last target location. If the user wants to move another document to the public space using the most recent setting, she simply needs to move in a straight line towards the public space. An empty segment is added to each ring allowing the user to keep whatever default value has been assigned to the object (functionally, this corresponds to moving through the empty space within a grid gate).

ATTRIBUTE GATES: SPATIAL CONFIGURATION

Selecting a sequence of attributes using a grid gate involves a number of targeting and steering operations. Figure 4 shows the steps required to select a sequence of gates. The worst-case scenario involves movement between attributes located at opposite extremes of the grid. Using the laws of targeting and steering derived from Fitts' law [1, 2, and 8], it is possible to optimize the grid layout.

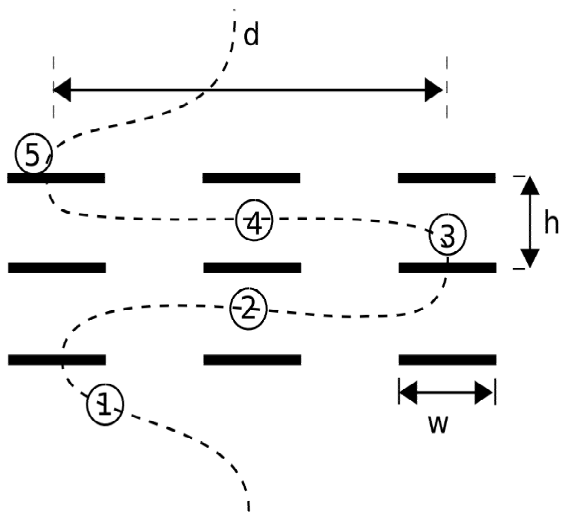

Figure 4: A worst-case scenario for setting a sequence of attributes using a grid gate.

The important parameters that determine the layout are the gate width (w) which is the target for Fitts' law, the row height (h) which is the path width for the steering law, and the maximum horizontal distance between the first and the last gate (d in figure 4) which is equal to 4w. It is possible to make use of similar calculations made by Dixon et al. [6] for optimising crossing based dialogs, indeed, one of their cases closely resembles moving through path 2 and crossing target 3 (figure 4). When the row height is small, the dominating factor becomes the steering index of difficulty:

$$ID_{steering} = \frac{path\,length}{path\,width} = \frac{4w}{h} \qquad (2)$$

Increasing the distance between rows eventually makes the targeting law the dominant factor, with an index of difficulty:

$$ID_{t\arg etting} = \log_2\left(\frac{A}{W}+1\right) = \log_2\left(\frac{\sqrt{h^2+(4w)^2}}{w}+1\right)$$
$$= \log_2\left(\sqrt{\left(\frac{h}{w}\right)^2+16}+1\right) \qquad (3)$$

Dixon et al. [6] found that a path width of 40px (for a target width of 18px, which in our case corresponds to a w/h ratio of about 0.5) is the width at which dominance switches between steering and targeting. Figure 5 shows a plot of the indices of difficulty for targeting and steering based on equations (2) and (3) for different w/h ratios. From the graph it is clear that an increasing (w/h) ratio decreases the targeting index of difficulty, and when the ratio exceeds about 0.25 further increases do not have a significant effect on the targeting index of difficulty. On the other hand the steering index of difficulty increases linearly with this ratio. Based on Dixon et al. [6], we find that when the w/h ration is about 0.5 the steering law starts to dominate, and steering starts to slow down the task. From this, we can conclude that a w/h ratio ranging between 0.25 to 0.5, in other words a row height twice to four times the gate width, should give the best performance.

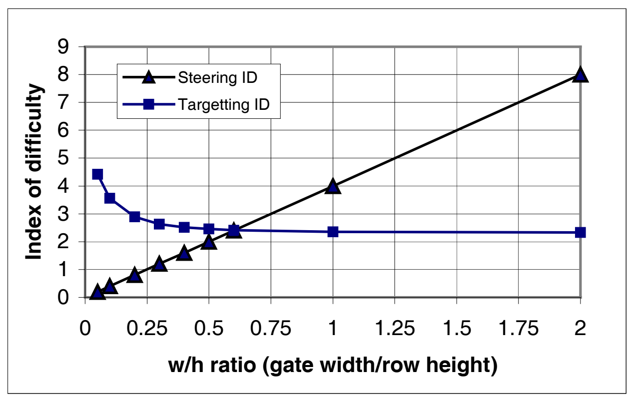

Figure 5: A graph showing the indexes of difficulty for steering and targeting.

We can use the same reasoning to select the layout parameters for polar gates. Selecting attributes using polar gates also involves a sequence of targeting and steering operations. The dominating factor for both targeting and steering is the width of the ring (w). When targeting, the ring width is much smaller than the gate width (figure 6) and is the dominant factor. Increasing w does not affect targeting as it increases the distance and the target width at the same time and hence only the effect of w on steering, and not targeting, needs to be considered. Increasing w increases the width for steering but at the same time increases the rings' circumferences and hence increases the distance. The steering law can be used to understand how w affects the steering index of difficulty.

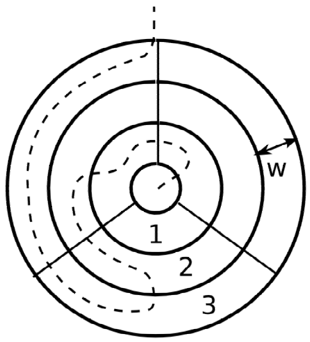

Figure 6: The path for setting a sequence of attributes using polar gates.

The distance for steering a full ring of index (i) from its center is:

$$d = 2\pi\left(r_0 + (i-1)w + \frac{w}{2}\right) \qquad (4)$$

Where r_0 is the inner radius for the first ring. The index of difficulty is given by:

$$\frac{d}{w} = 2\pi\frac{\left(r_0 + (i-1)w + \frac{w}{2}\right)}{w} = 2\pi\left(\frac{r_0}{w} + i - 0.5\right) \qquad \mathbf{(5)}$$

This shows that although increasing the width increases the distance, its over all effect is to reduce steering time. r_0 should also be made as small as possible.

In summary, for grid gates, it is better to make the distance between rows between two and four times the gate's width, and for polar gates it is better to keep the inner radius small and to use a large ring width (i.e., increase w in figure 6). In both cases these improvements increase the overall size of the gates, so there must be a limit on how large the gates should be. Accot and Zhai [2] have studied the impact of scale on steering tasks and their general conclusion was that best performance is achieved when the size is large enough to utilize all parts of the upper limb (i.e. arm, hand, and fingers), but not so large so as to place most of the effort required on the arms only, or so small as to place most of the effort required on the fingers only. With digital tabletops, unlike tablet PCs, it is possible to increase the size to a degree that utilizes the arm, hand and fingers. Increasing the size also increases awareness about the attributes being selected, but one should be careful not to make the gates too large that they slow down the performance as explained above.

The initial design for the gates was driven by aesthetic considerations. This involved using a narrow gap between rows and focusing on wide gates, similarly for polar gates, the inner area was made rather large, with narrower rings and a smaller overall size to reduce the gate's footprint. After applying the steering and targeting laws, it became clear that these were poor design decisions. Applying these guidelines, and doing some initial tests, demonstrated noticeable improvements in the usability of the gates.

RELATED WORK

Accot and Zhai [3] investigated crossing-based interfaces as an alternative to point-and-click interfaces, and showed that the time required for goal crossing was shorter, or at least the same, as for point-and-click. Accot and Zhai conducted their studies using a tablet input device connected to a desktop PC (with a standard display). In this setting the action and perception spaces are different, unlike in the case of digital tabletops and tablet PCs. Whether there is a difference between these two cases is an open question, but we assumed that when the two spaces coincide, a user's performance should improve or at least stay the same. Accot and Zhai suggested a number crossing-based interactions such as using directional goal crossing to turn a switch on or off depending on the crossing direction, performing double clicking by crossing a goal twice, or checking a number of options in one go if they are placed in a sequence (this is similar to the grid attribute gate).

Apitz and Guimbretière [4] built a crossing-based drawing application for tablet PCs named *CrossY*. *CrossY* introduced a number of new ideas on how to utilize this crossing technique in a complete application. Crossing was used to perform a range of operations including setting the pen width and color, find and replace, scrolling, and using the motion of setting a number of attributes in one go by passing over them in one stroke.

Although polar gates are not contextual menus and can be thought of as equivalent to a dialog box of related on/off commands, the concept of showing commands in a polar layout and activating them by passing over them is not completely new. A number of polar menus have been suggested including Pie menus [10], Marking menus [12], Control menus [17], and FlowMenu [9]. FlowMenu was used with CrossY and was also used by Scott [19] with digital tabletops. Particularly related ideas in FlowMenu include the crossing-based interface, the polar layout, and that it is possible to make a number of consecutive menu selections without the pen having to leave the active surface. However, if we imagine using FlowMenu to set the scale, rotation, and access right attributes, we would still need to use the menu three times, once for each attribute.

EVALUATION

We designed a user study to compare ease of use, performance, and mutual awareness for the two types of gates, with standard contextual menus on a pen-based digital tabletop. We used a pre-production prototype of the multi-pen Promethean Activboard (figure 7). As with other studies of tabletop interface elements we used a mixture of precision and timing measures, as well as reflection by our participants, both to document users' experience of attribute gates and to contrast their use with standard interface elements.

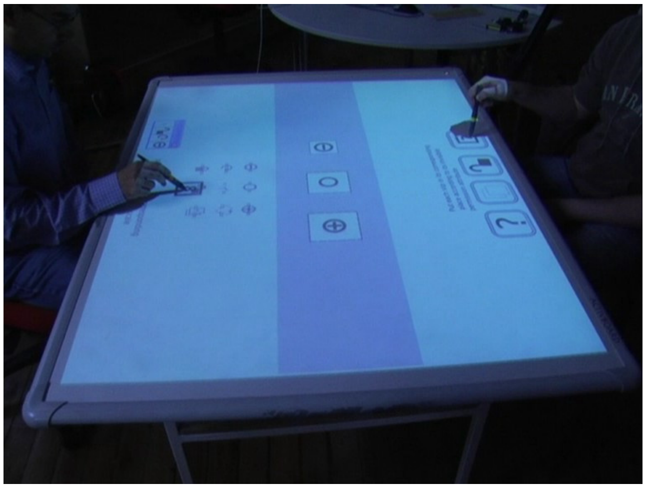

Figure 7: Configuration of the multi-pen surface and the two participants for the user study. The sender is on the left, and the receiver is on the right. The small dialog box to the left of the sender (which was shielded from the receiver) shows the required settings for the current object.

The application designed for the study involved two users, the *sender* (the user on the left in figure 7) and the *receiver* (the user on the right in figure 7). To the left of the sender, the application provided a prompt as to the attributes to be applied to the current object. This involved setting three values for each of the following attributes:

- access rights (read-only/read-write/duplicate)
- orientation (manual/to-centre/rotate-and-translate)
- scale (enlarge/no-change/shrink).

The sender's task was to apply these settings to the current object, and place the object in one of three locations in a public space in the centre of the table according to the value of the scale attribute. The gates and the contextual menu appeared when the sender pressed and held the pen tip on the object for half-a-second. The gates disappeared when the pen tip was released. The contextual menu disappeared after the selection of each subcommand.

Orientation and scale settings have a direct effect on the appearance of the object, unlike the access rights settings. That is why the receiver's task was to monitor the value of the access rights attribute applied by the sender, and to move the object from the shared space to a second set of locations (in her personal space) according to the observed value of this attribute. With attribute gates the access rights attribute set by the sender stays visible as long as the gate is visible, unlike the case with contextual menus where the menu disappears after clicking on the required setting. As can be seen in figure 7, in front of the receiver are four square-shaped areas. Three of these correspond to the three access rights settings available for the sender, and a question mark labels the fourth area. Receivers were asked not to guess the value of an attribute. In cases where they were unsure of the value they should put the object in the area labeled with the question mark. A barrier was placed on the tabletop to prevent the receiver from seeing the access right setting prompt provided to the sender (not shown in figure 7). Although senders and receivers worked in parallel, they were asked neither to support nor hinder each other.

Each sender/receiver pair was required to perform the experiment on three sets of 10 objects, using a different attribute assignment interaction technique for each set (grid gate, polar gate, and contextual menu). The actions required by a sender depended on the attribute assignment. That is, for grid and polar gates, different attribute value combinations require different ranges of movement. Consequently, the sequence of attribute values to be assigned was repeated for each interaction technique. Three of the setting combinations (for object 2, 6 and 7) were deliberately set to values that were the same as their predecessor. This allowed us to explore the potential benefit of the self-configuring characteristic of the polar gates.

Eight sender-receiver pairs of experienced computer users took part in the study. The pairs were allowed to practice the task using all three interaction techniques without a time limit. When they felt confident with all the techniques they commenced the study (practice times varied between 4 and 10 minutes). The training and trial sessions were both supervised. We measured the following properties for each trial (i.e. for each assignment of attribute values):

1. time taken for the sender to apply the setting, measured from the initial selection of the object to the placement of the object in the central public space;
2. sender's accuracy in assigning the attribute values;
3. accuracy of the receiver's judgment as to the value of the access right attribute (the value actually set by the sender).

Accuracy

Results of the study show that the senders accurately assigned the attributes using all the techniques, with only one user making a single erroneous assignment.

Awareness

The attribute gates showed apparent advantages over the contextual menu with regard to the receiver's awareness of the attribute assignment. Figure 8 shows the distribution of errors across the participants for each interaction technique. Only one receiver made one error under the attribute gate conditions. By contrast, in the contextual menu condition nearly 19% of trials resulted in either an incorrect judgment by receivers, or the receivers indicated that they were unsure of the access right attribute value. These cases were distributed across 6 of the 8 receivers, which indicates that it is related to some aspect of the interaction technique (rather than the participants themselves).

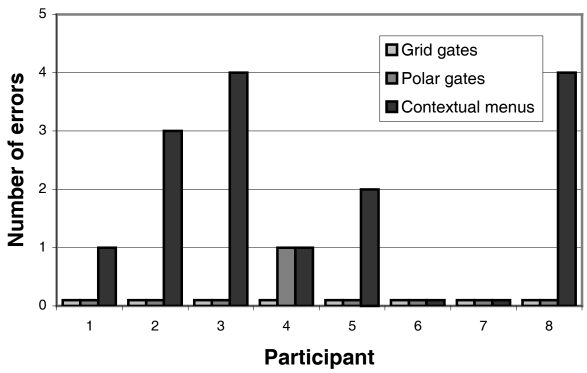

Figure 8: Receiver errors (by participant) for each interaction technique.

Performance

An analysis of variance showed that participants performed the sender's task significantly faster using attribute gates (see figure 9). The average time in seconds for completing the task for each type were: grid gates=5.9, polar gates = 7, and contextual menus=13.1. Comparing between grid and contextual menus showed $F_{1,18}$=67.9, p=1.60E-07; and between polar and contextual menus $F_{1,18}$=32.3, p=2.15E-05.

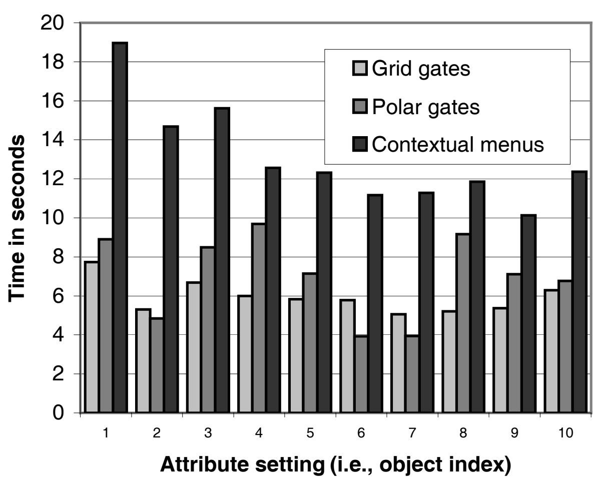

Figure 9: Average timings for the sender task for each attribute setting (by interaction technique). Each bar represents the average time of the eight participants for setting the indicated object.

There was not a significant difference between the polar and grid gates ($F_{1,18}$=2.2, p=0.16) although grid gates led to slightly better average performance than the polar gates, except in the case of the repeated attribute assignments (Settings 2, 6 and 7 in Figure 9) for which the participant's use of the polar gates was slightly faster. Figure 10 shows the average time of each participant using each type. This clearly shows that the performance of participants when using the contextual menus was consistently worse.

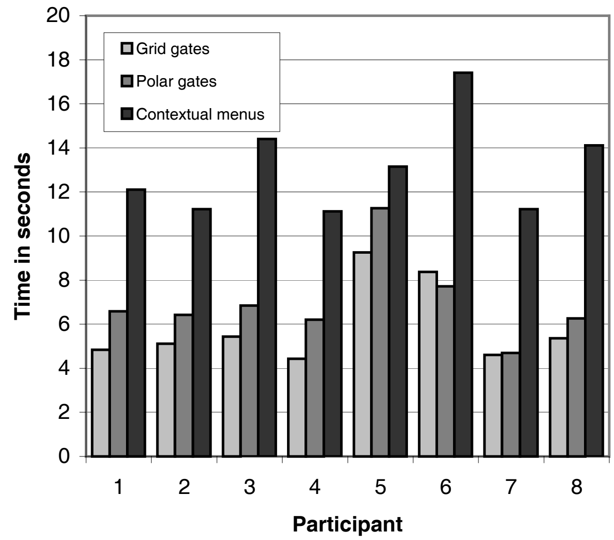

Figure 10: Average timings for each participant (by interaction technique). Each bar represents the average time for setting all the objects for one participant using one of the techniques.

Participant views

Participants were interviewed after the study and asked for their comments and preferences. Despite the relatively poor performance of participants with the contextual menus, two participants expressed a preference for these, including one participant whose performance with the attribute gates was faster than any other participant. Most of the participants, however, preferred attribute gates with varying preferences between the grid and polar layouts. A number of participants identified that the projection from above, and the resulting obscuration of attributes made the polar gates harder to use.

DISCUSSION

Our two configurations of attribute gates were developed in response to our analysis of the requirements of an interaction technique to integrate attribute setting with the movement of objects between territories in tabletop interfaces. Attribute gates provide a fluid way to change a number of different settings when moving objects between territories on the table. Attribute gates are motivated by the principles of task levels in activity theory and crossing-based interfaces. The spatial configuration of attribute gates was optimized using targeting and steering laws derived from Fitts' law.

Both polar and grid gates are crossing-based interfaces that allow users to concurrently move interface objects and set their attributes in one smooth action. Conventional design wisdom for tabletop interfaces would suggest that, in the absence of readily accessible toolbars and system menus, contextual menus should be used to set object attributes. An evaluation of both forms of attribute gate demonstrated significant advantages over standard contextual menus in terms of user performance and mutual awareness.

Polar gates and grid gates differ in a number of ways. Firstly, grid gates maintain the spatial location of the gates themselves. Although not examined in our study, a prolonged evaluation may demonstrate an additional benefit as users internalize these positions and use more automatic free flowing strokes. While faster setting of attributes may impinge on mutual awareness, bystanders will gain similar familiarity with these positions and movements.

Polar gates, deliberately maintain the configuration of "last use" with a view to exploiting the fact that attribute combinations are often repeated for a specific user engaged in a particular task. The benefit of this persistence of state was demonstrated in the user study, for which the polar gates slightly outperformed the grid gates for repeated states, without impacting on mutual awareness.

A final observation is that the polar gate's use of a physical metaphor (manipulating the concentric rings) encouraged users' steering behavior. In simple terms, the mechanism of the polar gate was more readily understood by users. Though the metaphor of the grid gate was clear, the freedom of movement afforded between gates resulted in them requiring more practice time than for the polar gates. Another important advantage of polar gates is that they are orientation independent. This is an important feature characterizing components specifically designed for tabletops where users around the table view components from different angles.

Although designed for the specific problem of setting access rights, orientation and scale when moving objects between territories [16, 19], attribute gates have broader applicability. Attribute gates in essence integrate the multidimensional (in attribute space) character of a dialogue box with the fluidity of pen-based interaction.

ACKNOWLEDGMENTS

Thanks to Promethean Limited for the loan of the multipen Activboards and Diwan Software Limited for financially supporting this research.

REFERENCES

1. Accot, J. and Zhai, S. Beyond Fitts' law: models for trajectory-based HCI tasks. *Proceedings of the SIGCHI conference on Human factors in computing systems,* p.295-302, March 22-27, 1997.

2. Accot, J. and Zhai, S. Scale effects in steering law tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, United States). CHI '01. ACM, New York, NY, 1-8.

3. Accot, J. and Zhai, S. More than dotting the i's --- foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 73-80.

4. Apitz, G. and Guimbretière, F. CrossY: a crossing-based drawing application. In *Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology* (Santa Fe, NM, USA, October 24 - 27, 2004). UIST '04. ACM, New York, NY, 3-12.

5. Buxton, W. A. Chunking and phrasing and the design of human-computer dialogues. In *Human-Computer interaction: Toward the Year 2000*, R. M. Baecker, J. Grudin, W. A. Buxton, and S. Greenberg, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 494-499. 1995.

6. Dixon, M., Guimbretière, F., and Chen, N. 2008. Optimal parameters for efficient crossing-based dialog boxes. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1623-1632.

7. Drury, C. G. Movements with lateral constraint. Ergonomics, 14(2), 1971. p. 293-305.

8. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47. 1954. p 381-391.

9. Guimbretiére, F. and Winograd, T. FlowMenu: combining command, text, and data entry. In *Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology* (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM, New York, NY, 213-216.

10. Hopkins, D. The Design and Implementation of Pie-Menus. *Dr. Dobb's Journal*. 1991. 16(12): p. 16 - 26.

11. Kaptelinin, V. Activity theory: implications for human-computer interaction. In *Context and Consciousness: Activity theory and Human-Computer interaction*. B. A. Nardi, Ed. Massachusetts Institute of Technology, Cambridge, MA, 103-116. 1995.

12. Kurtenbach, G. P. *The Design and Evaluation of Marking Menus*. Doctoral Thesis. UMI Order No. GAXNN-82896., University of Toronto. 1993.

13. Kuutti, K. Activity theory as a potential framework for human-computer interaction research. In *Context and consciousness: activity theory and human-computer interaction.* B. A. Nardi, Ed. Massachusetts Institute of Technology, Cambridge, MA, 103-116. 1995.

14. Kruger, R., Carpendale, S., Scott, S. D., and Greenberg, S. Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication. *Comput. Supported Coop. Work* 13, 5-6 (Dec. 2004), 501-537.

15. Kruger, R., Carpendale, S., Scott, S. D., and Tang, A. Fluid integration of rotation and translation. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA,

April 02 - 07, 2005). CHI '05. ACM, New York, NY, 601-610.

16. Morris, M. R. *Supporting Effective Interaction with Tabletop Groupware*. Ph.D. dissertation, Dept. of Computer Science, Stanford University, 2006.
17. Pook, S., Lecolinet, E., Vaysseix, G., and Barillot, E. Control menus: excecution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 263-264.
18. Ringel, M., Ryall, K., Shen, C., Forlines, C., and Vernier, F. Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1441-1444.
19. Scott, S. D. Territory-Based Interaction Techniques for Tabletop Collaboration. *Conference Companion of the ACM Symposium on User Interface Software and Technology.* UIST 2003.
20. Shen, C., Everitt, K., and Ryall, K. UbiTable: Impromptu Face-to-Face Collaboration on Horizontal Interactive Surfaces. *TR-2003-49, Mitsubishi Electric Research laboratories* (2003).
21. Vernier, F, Lesh, N., Shen, C. Visualization Techniques for Circular Tabletop Interfaces. *Proc. AVI* 2002, 257-263.
22. Winograd, T. and Guimbretière, F. Visual instruments for an interactive mural. In *CHI '99 Extended Abstracts on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, May 15 - 20, 1999). CHI '99. ACM, New York, NY, 234-235.