Rotation and Translation Mechanisms for Tabletop Interaction

Mark S. Hancock[1,2], Frédéric D. Vernier[1,3], Daniel Wigdor[1,4], Sheelagh Carpendale[2], Chia Shen[1]

[1]Mitsubishi Electric
Research Labs
201 Broadway
Cambridge, MA, USA
02139

[2]Interactions Lab
Dept. of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, AB, Canada
T2N 1N4

[3]University of Paris XI
LIMSI-CNRS Lab
BP133,
91403 Orsay, France

[4]DGP Lab
Dept. of Computer Science
University of Toronto
10 King's College Rd.
Toronto, ON, Canada,
M5S 3G4

shen@merl.com   {msh,sheelagh}@cs.ucalgary.ca   frederic.vernier@limsi.fr   dwigdor@dgp.toronto.edu

Abstract

A digital tabletop, such as the one shown in Figure 1, offers several advantages over other groupware form factors for collaborative applications. However, users of a tabletop system do not share a common perspective for the display of information: what is presented right-side-up to one participant is upside-down for another. In this paper, we survey five different rotation and translation techniques for objects displayed on a direct-touch digital tabletop display. We analyze their suitability for interactive tabletops in light of their respective input and output degrees of freedom, as well as the precision and completeness provided by each. We describe various tradeoffs that arise when considering which, when and where each of these techniques might be most useful.

1. Introduction

Direct-touch tabletop systems, such as the one shown in Figure 1, offer several advantages over other groupware topologies. Among these are the opportunity to sit face-to-face with collaborators, and that participants are all given equal control of on-screen objects, without the need to take turns with an input device. Unlike other topologies for collaborative groupware, users of a tabletop system do not share a common perspective for the display of information: what is presented right-side-up to one participant is upside-down for another.

The orientation of text has a significant effect on reading performance [7]. Researchers have also found that, when collaborating around a traditional meeting-table, the position and orientation of objects is used as semantic information, such as for grouping of objects or to denote ownership; subtle variations in rotation and position can serve as a language to communicate intention within and among group members [8,14]. A challenge for researchers of tabletop groupware,

therefore, is how to allow objects in such a system to be dynamically reoriented by the participants in such a way as to allow the full expressiveness of this language.

Graphical programming tools, such as Java 2D™, OpenGL®, and DirectX®, allow for dynamic reorientation and positioning of on-screen objects. Not at all addressed by these technologies, however, is the design of an interface for the user to specify these parameters. Several techniques have been developed by a multitude of researchers in an attempt to address this problem, each with its own advantages and disadvantages. What has become clear is that a tension may exist between two important aspects of such a technique: leveraging the power of technology to provide precise control of these parameters while maintaining the natural interaction familiar in our use of traditional tables and paper.

The contributions of this paper are threefold. First, we provide a survey of existing rotation and translation mechanisms available for interactive tabletops using a language of common variables to describe them.

Figure 1. A multi-user, direct-touch table system. Each participant has a unique perspective, necessitating dynamic reorientation of objects.

Second, we provide a means to compare and contrast these various techniques through consideration of input to and output from these algorithms. Third, we provide a discussion of the strengths and weaknesses of each technique, as well as design considerations for future interactive tabletop designers.

2. Methods of Rotation & Translation

In order to better compare the various methods of rotation and translation, we define several common variables. The following points are used as input to the algorithms used in the interaction techniques for rotation and/or translation of objects on a direct-touch input device (as illustrated in Figure 2):

T – the point of initial input contact (for multiple points of contact the notation T_1, T_2, etc. is used)

T' – the point at which the input contact is released (similarly, T'_1, T'_2, etc. are used)

O – the center of the table (display)

C – the center of the object before it is moved and/or rotated

C' – the center of the object after it has been moved and/or rotated

For ease of comparison, the properties of the different rotation and translation algorithms that are discussed in this section have been summarized in Table 1. The common properties of any rotation and/or translation algorithm are the *center of rotation* which is typically one of T, O or C, one or more *fixed points* on the object that remain independent of the rotation (i.e. points that only translate), and the *angle of rotation* (θ).
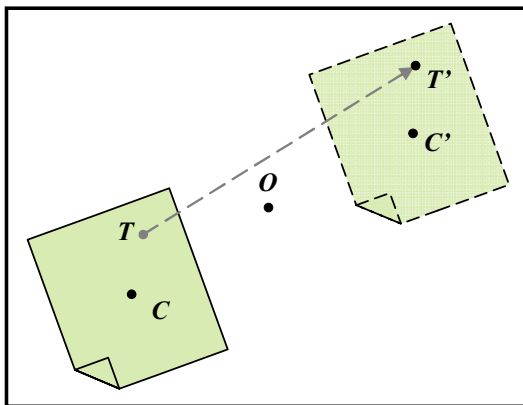
Figure 2. Input parameters of rotation and translation. This diagram shows an example of translate-only motion.

	Fixed Point	Center of Rotation	Angle of Rotation
Independent (Translation)	All points	N/A	N/A
Independent (Rotation)	$C \rightarrow C$	C	$\angle TCT'$
Automatic (Continuous)	$T \rightarrow T'$	O	$\angle TOT'$
Automatic (Discrete)	$T \rightarrow T'$	O	$2\pi/n$
Integral (RNT)	$T \rightarrow T'$	T	$\angle TCT'$
Two-point	$T_1 \rightarrow T'_1$ or $T_2 \rightarrow T'_2$	$T_1, T_2,$ or Midpoint	$\angle T_2 T'_1 T'_2$

Table 1. Comparison of the various rotation and translation methods.

2.1. Explicit Specification

Users can invoke an explicit command to perform either a relative change in or absolute specification of position and orientation. For tabletops, Wu and Balakrishnan [19] demonstrate the use of a rotation widget, using a two-point gesture with the thumb and index finger, with which a user can adjust an object's orientation by manipulating the three digits of the rotation angle (see Figure 3).

In general, users can provide this explicit command through a variety of input techniques, including menu selections, gestures, and the use of keyboards or numeric keypads. This method is typical in many drawing application, Users can provide this explicit command through a variety of input techniques, including menu selections, gestures, and the use of

Figure 3. A widget to manually specify the desired orientation of an on-screen object ([19]).
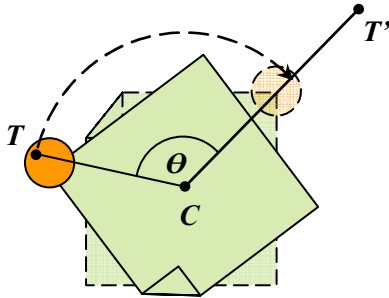
Figure 4. The most commonly used method of rotation is an independent rotation that occurs when the initial contact point (*T*) is within a specified area. The rotation is performed about the center, *C*, at the angle *θ*.

keyboards or numeric keypads. This method is typical in many drawing application, including Microsoft© Paint and the Gnu Image Manipulation Program (GIMP).

2.2. Independent Rotation & Translation

Another method to control the rotation and translation of an object on the screen is to provide designated areas within which the user can control either rotation or translation. Thus, if *T* is inside a translation-only area, the object is translated by the difference between *T* and *T'* and if *T* is inside a rotation-only area, the object is rotated about *C* by the angle *θ*, formed by *T*, *C*, and *T'* (see Figure 4). Note that the dedicated area can be the entire object, in which case rotation vs. translation can be specified via a mode. This method is used in many graphics applications for the desktop computer, such as GIMP, Microsoft Powerpoint®, and 3D Studio Max®[1] and in many tabletop display systems [11,12,13].

2.3. Automatic Orientation

Rotation and translation can be combined into one motion. One method of combination is to automatically rotate the object to face a system-determined direction as it is moved. The system can determine the direction of rotation based on location of other objects on the screen, such as snapping [1], or based on the location of the users at the table.

The location of the users can be obtained either automatically [2] or (more simply) based on the assumption that the user is at the outside border of the table [11,16]. With the latter assumption, the system

---

[1] In 3D Studio Max, this mechanism is further separated into rotation/translation in separate planes in three dimensions.

can automatically orient an object so that its top always faces the center of the table (*O*) and its bottom faces the outside border of the table, therefore it is readable by a user at that particular border (see Figures 5 and 6). This orientation is easily calculated from the angle component of an object's position in polar coordinates. Because of this simplicity, the resulting interaction is predictable and may, therefore, be understandable to users. However, this simplicity has the consequence that not all rotation/position combinations can be achieved (e.g., only polar translations). Therefore when the underlying assumption is violated (e.g., a user at a far edge desires readability of an object without translating it), this technique will need to be augmented with some other mechanism, such as rotation-only corners, in order to provide arbitrary object orientations.

Many variations of this table-centric technique have been investigated [3,11,16]. In a *continuous* variation, the object is rotated about *O* by the angle *θ*, formed by *T*, *O*, and *T'* and translated along the axis *OT'* by the difference between |*OT*| and |*OT'*|. Thus, the point *T* is a fixed point on the object and is translated to point *T'* (see Figure 5).

In a *discrete* variation, the interaction is similar, except that the rotation occurs when (and only when) the object is moved across a predefined boundary. It is simplest when these boundaries are straight lines from the center (*O*) to an edge of the table and occur at regular intervals. This regular setting can be expressed mathematically as:
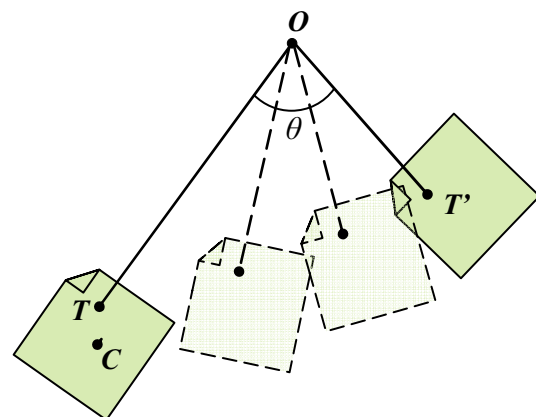

Figure 5. In *continuous* automatic rotation, the object is translated so that the initial touch point, *T*, is moved to the new touch point, *T'*, but the angle of rotation is relative to the center of the table, *O*.
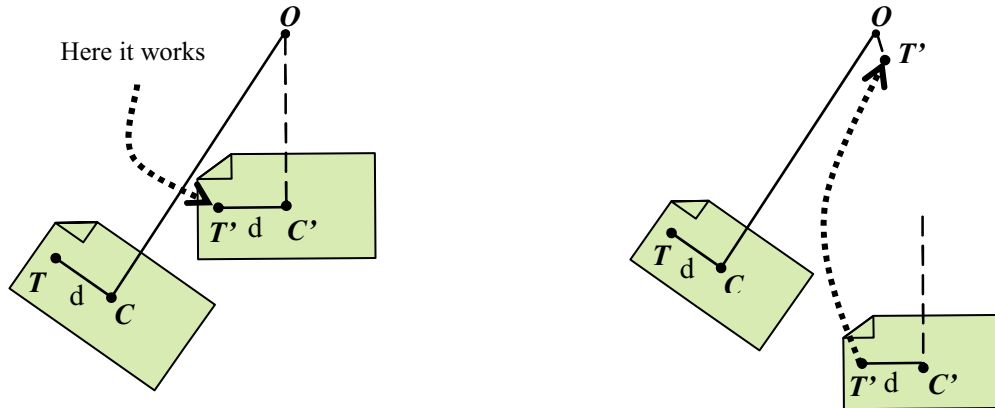
Figure 6. *C* **can be used as a fixed point, which works nicely for movement far from** *O* **(left), however, a problem occurs when the distance between** *T* **and** *C* **is less than the distance between** *T'* **and** *O* **(right). This problem can be solved by relaxing the constraint that** *T* **remain under the user's finger, or by initially snapping** *C* **to** *T* **when the movement starts.**

$$\theta = \begin{cases} -\dfrac{2\pi}{n}, & \text{if left boundary crossed} \\ 0, & \text{if no boundary crossed} \\ \dfrac{2\pi}{n}, & \text{if right boundary crossed} \end{cases}$$

where *n* is the number of boundaries.

For example, if *n*=4, then an object facing an outer edge of a four-sided table will rotate by 90º when it approaches an adjacent edge of the table to face the new edge. Transitions across the boundary can be made smooth through animation. This method is also referred to as Table For 2, Table For 4, or Table For N [11].

With all table-centric techniques, the interaction is somewhat problematic when *T* is close to *O*, since large changes can occur suddenly. To prevent these large changes, *C* and *C'* can be used in place of *T* and *T'*. However, this requires either relaxing the condition that *T* remain fixed, or initially snapping *T* to *C* when the interaction begins in order to prevent undefined behavior when *O* is within the triangle *TCT'* (see Figure 6).

2.4. Integral Rotation & Translation

Another way to combine rotation and translation is to use a physically-based model to simulate integrated rotation and translation that occurs with real paper on the surface of a table. Kruger et al. [9], Mitchell [10] and Beaudouin-Lafon [1] present different techniques of integrated translation and rotation that introduce a frictional element to the movement of objects on the tabletop display. The methods used by Kruger et al. and Beaudouin-Lafon., named RNT (Rotate N' Translate) in the former, are mathematically

equivalent. In both RNT and Mitchell's methods, the amount of rotation is relative to the component of motion *TT'* that is perpendicular to *CT*. Both methods also have the property that if *TT'* is in the same direction as *CT*, then the object is translated and not rotated. These two methods differ in that the former method causes a 180º rotation when *TT'* is in the opposite direction as *CT*, but the latter causes no rotation.

More specifically, the RNT algorithm causes a translation by the difference between *T* and *T'* and a rotation about *T* by the angle θ, formed by *T*, *C*, and *T'* (see Figure 7). This algorithm is executed at each frame of motion, thus two identical paths can result in different rotations depending on the frame rate and/or speed of motion.

2.5. Two-Point Rotation & Translation

Rotation and translation can also be integrated using two points of contact. The translation can be
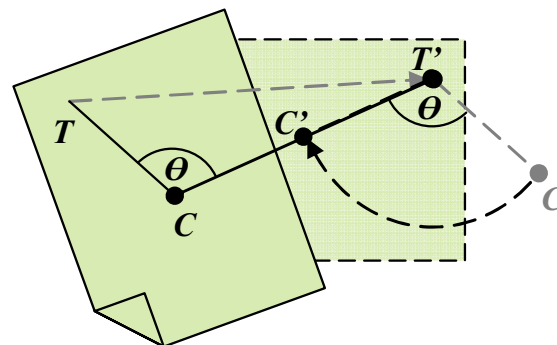

Figure 7. The RNT algorithm uses a physically-based model to calculate the rotation at each frame. The object acts as if a force of friction is being applied opposite the direction of movement.
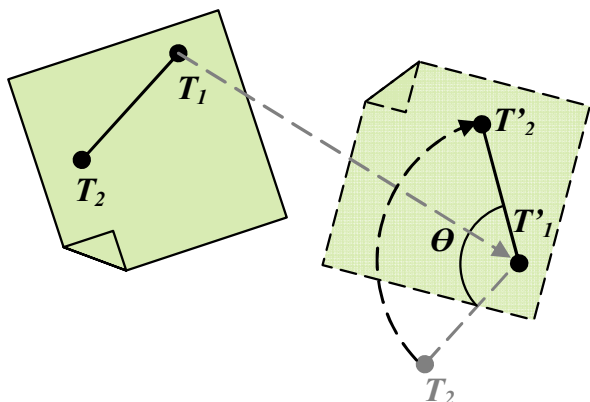
Figure 8. In two-point rotation, the first contact point, T_1, is used both as the center of rotation and as a fixed point in translation. The object is rotated based on the angle $\angle T_2 T'_1 T'_2$.

determined exclusively by the difference between T_1 and T'_1 for the first point of contact and the object can be rotated about T'_1 by the angle θ, formed by T_2, T'_1, and T'_2 (see Figure 8). This second point of contact can also be used to simultaneously resize the object [2] by a scale factor of $|T'_1 T'_2|:|T_1 T_2|$. If this scaling is not done, however, the user must maintain a constant distance between T_1 and T_2 in order to maintain the constancy of the second touch point. Maintaining a constant distance between the points while moving is more difficult on a tabletop display than with paper, since there is no physical tension between the fingers to help maintain their relative positions. Alternative fixed points can be used along $T_1 T_2$ (this mid-point is a likely candidate), or priority can be given to one of the two fingers (see Table 1), but these all have a similar disadvantage.

3. Comparisons and Design Guidelines

To compare the various techniques, it is useful to consider the degrees of freedom (DOF) of input and output (see Table 2). For input, *explicit specification* uses 1DOF, *independent rotation & translation*, *automatic orientation* and *integral rotation & translation* all use 2DOF, and *two-point rotation & translation* uses 4DOF. For output, *explicit specification* and *independent rotation* use 1DOF, *automatic orientation* and *independent translation* use 2DOF, *integral rotation & translation* uses 3DOF, and *two-point rotation & translation* uses 3 or 4 DOF.

Consideration of the mapping between input DOF and output DOF is illustrative. The one-to-one mapping in independent translation provides a clear mental mapping for the user. However, in independent rotation, the projection of two dimensions onto one

may be confusing (e.g., why wouldn't the point of contact remain under a user's finger?). The one-to-one mapping for automatic orientation alleviates this confusion, however, the technique is limited to 2DOF output and so is incapable of certain position/angle combinations (as is translation-only). Integral rotation and translation provides a method of specifying all 3DOF from 2DOF input, and so can specify an arbitrary position/angle combination with one motion in two dimensions. However, because of the increase in dimension, some other variable, such as direction of motion must be used, and thus the technique is dependent on change in this variable over time (i.e. speed of motion). The mapping of 4DOF to 3DOF in two-point interaction suffers the same problem as independent rotation (e.g., why doesn't the second point of contact remain under the user's finger?) and is only alleviated by mapping the fourth dimension to another variable (e.g. size).

A change in DOF between input and output can be both beneficial and costly, depending on the intended goal of the interaction. When the DOF of output are no more than the DOF of input, it may be easier for the user to precisely specify the output parameters. However, increasing the DOF of output may provide a means to perform a complex task in one motion, thus providing a more quick and natural interaction.

3.1. Coordination and Communication

Studies [14] have shown that users tend to use the orientation of objects to communicate to other users in a co-located environment. Subtle changes in rotation of an object can indicate a user's intention to share information, to indicate territoriality, or to invite participation. In this sense, the position and angle of an

	Input	Output
Explicit Specification	x or y or θ	x or y or θ
Independent Rotation	x and y	θ
Independent Translation	x and y	x and y
Automatic (table-centric) Orientation	x and y	r and θ
Integral Rotation & Translation	x and y	x, y, and θ
Two-Point Rotation & Translation	x_1, y_1, x_2, y_2	x, y, θ, and (*size*)

Table 2. Degrees of freedom of input and output for each rotation & translation technique. x and y are the two coordinates of T, with subscripts to identify multiple touch point, as before.

object together serve as a type of language or form of expression between group members. In order to facilitate this communication, an interaction technique must be natural enough so as not to interfere with this intention. Thus, if the intention of the interaction is to provide a natural mechanism for the expression of this language, the designer should choose a technique that supports 3DOF of output, even when only 2DOF of input are available.

Furthermore, it has been shown that rotation and translation are perceived as integral and not separable in the human mind [5, 17], and so it is appropriate to use an integral technique, such as RNT or two-point interaction, to support this expectation.

3.2. Consistency

Although integral techniques provide a mechanism consistent with the human mental model, providing a higher DOF of output than input introduces a problem of consistency. Such a technique must use some other variable to determine the extra output variable and so the input DOF alone does not determine the output. For instance, the RNT technique has the property that two paths that have the same start and end points may not have the same resulting position and angle. When the output DOF is no more than the input DOF, the input parameters completely specify the position and/or orientation of an object. For example, when using automatic table-centric orientation, no matter what path a user moves a document, the final position will indicate the translation and rotation based on the polar coordinates (r and θ) of the object.

3.3. Completeness

By providing the user with a mechanism to rotate and translate an object, it is desirable that any position and angle be obtainable. The DOF indicate whether or not all possibilities can be achieved in one motion. Techniques with 3DOF output, such as RNT and two-point rotation and translation, provide completeness with one interaction motion. Techniques with 2DOF or less of output must be combined with another technique to provide the other DOF, such as automatic orientation and independent rotation and translation.

3.4. GUI Integration

In a traditional desktop graphical user interface (GUI), windows, icons, menus and pointers (WIMP) are typically constrained to a single angle, due to the use of a vertical display. Thus, these objects are easily manipulated using a translation-only interaction technique, with 2DOF input and output. Because the interaction is uniquely specified by its input DOF, an application can limit the interaction to a limited portion of an object, such as the title bar. Use of this visually maintained mode (when the mouse cursor is over the title bar) allows for other modes of interaction, such as text selection or navigation control, in the remaining portion of an object.

The use of visually maintained modes in a tabletop display environment can also be made possible. However, like in desktop interfaces, it is beneficial to use a technique that matches the input DOF to the output DOF. The automatic table-centric orientation provides a good alternative to the translate-only mechanism in desktop applications for a tabletop display. Because it maps 2DOF onto 2DOF, the title bar of an object can easily be used to provide a mechanism of rotation and translation on a table.

Using a higher DOF of output than input requires a larger portion of the object to be dedicated to rotation and translation. For example, RNT requires the use of the entire boundary to specify an arbitrary rotation and translation. Using a lower DOF of output than input makes it difficult to constrain the interaction within a small area. For example, using rotation-only corners or two-point rotation and translation allows one of the points of contact to exit the designated area during the operation.

3.5. The Role of Snapping

Snapping of objects can be thought of as a reduction in DOF from input to output. Thus, independent rotation and two-point rotation and translation (without resizing) can be considered a form of snapping. However, snapping can also be integrated into any of the other techniques. For example, when an object is moved to the edge of the table, the motion can be constrained so that the object only moves in 1DOF along that edge.

This reduction in DOF can be beneficial in the support of precise interaction. Much like a ruler can facilitate the straightness of line drawing, snapping can be used to specify precise interactions in many ways on a tabletop display.

Snapping of objects to an unseen grid, or to one another, is a common technique for helping to align objects in an application. In [1], objects are made to snap orientationally, and in [4] 3D objects can be made to snap to one another both orientationally and positionally. This snapping behavior simplifies the task of aligning objects to one another, or to some absolute grid, either of which may be appropriate for any given application.

There are several types of snapping which might be included in a system that implements any of the techniques we have described:

- Snap to a polar-grid
- Snap to a rectilinear grid; or
- Snap objects to one another

In all cases, the snapping may constitute a snap of position, orientation, or both.

4. Concluding Remarks and Future Work

Ideally, tabletop display interfaces should support ease of interaction, communication, coordination, consistency, completeness, GUI integration, and precise snapping. However, it seems clear that no one interaction technique is suitable for all applications. For instance, when users are performing highly collaborative work and must coordinate their actions, it is important to support natural interactions. In this case, it may be best to use an interaction technique such as RNT or automatic orientation (depending on the type of collaborative activities and whether GUI integration is required). On the other hand, when a group (perhaps the same one) is performing precise actions, such as final touches, it may be best to use more precise techniques, such as independent rotation and translation together with snapping.

Because coordination requires a high DOF output while precise interaction requires a reduction in DOF output, these may be conflicting design goals. However, users may need the freedom of natural interaction to allow for easy communication around the table, but the accuracy of precise interaction to complete a task. For instance, in a group of users performing a layout task for a newspaper, one user may wish to discuss the use of a photograph with her colleague by turning the photograph toward him. Once they agree on its inclusion, she may then want to align it within its corresponding text, parallel to the edge of the current layout. Thus, the appropriate choice of rotation and translation algorithm may be a mixture of those suggested in this paper, one for the natural communication task and the other for the precise alignment task. The consideration of seamless transition between the techniques would be essential to any successful design choice.

We suggest that the challenge for future designers of tabletop interfaces is not to make any one particular choice of which interaction technique to include in their application, but rather how to provide a seamless transition between and amongst appropriate techniques.

In this paper, we have analyzed in depth the properties of five existing document rotation and translation techniques, and presented a set of comparative design guidelines. In an effort to fully understand the naturalness and ease of use of these interaction techniques, we have also implemented a subset of these techniques, namely the automatic orientation augmented with independent orientation and translation schemes, as well as a separate RNT, within the ongoing DiamondSpin toolkit [11].

At present, we are continuing the user evaluation of some of the rotation and translation techniques, in order to evaluate their true interaction experience to the users, as well as their performance implications, compared with their theoretical properties presented in this paper. Our initial findings in [3] suggest that a more objectively precise technique may not match the more qualitative ratings of user preference as each technique induces a different "feel" to the user. This different feel can be expressed in terms of *interaction fluidity*, *learnability* of how a technique behaves under the user's touch, and the perceived *naturalness* of a particular technique. This is consistent with findings in [9] which suggest that a user's perception of precision may be related to the time spent completing the interaction, thus causing natural techniques to, deceptively, seem less precise. Further studies are required to obtain a comprehensive understanding of the implications of these tabletop rotation and translation techniques.

Multi-user collaborative interactive tabletop displays are still in their infancy. Thus, applications for digital tabletops are still evolving, and it is of utmost importance to systematically study the implications of the interaction techniques we bring to bear within this young field. The work presented in this paper is intended to contribute to our collective knowledge and to aid in the design of usable computationally augmented tabletops.

5. References

[1] Beaudouin-Lafon M. (2001) 'Novel Interaction Techniques for Overlapping Windows', Proceedings of the UIST'01 conference, ACM Press, pp. 153-154.

[2] Fitzmaurice, G. W., Balakrishnan, R., Kurtenbach, G., and Buxton, W. (1999) 'An exploration into supporting artwork orientation in the user interface', In Proceedings of ACM CHI'99 Conference on Human Factors in Computing Systems, pp. 167–174.

[3] Forlines, C., Shen, C., Vernier, F., Wu, M. (2005) 'Under My Finger: Human Factors in Pushing and Rotating Documents Across the Table', Tenth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2005), (in press).

IEEE
COMPUTER
SOCIETY

[4] Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., Buxton, B. (2001) 'Interaction techniques for 3D modeling on large displays', In Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 17-23.

[5] Jacob, R. J. K., Sibert, L. E., McFarlane, D. C., Preston Mullen, J. M. (1994) 'Integrality and separability of input devices'. ACM Transactions on Computer-Human Interaction (TOCHI) 1 (1), pp. 3–26.

[6] Hancock, M.S., Booth, K.S. (2004). 'Improving menu placement strategies for pen input', Proceedings of the 2004 conference on Graphics interface, ACM Press, pp. 221-230.

[7] Koriat, A. and Norman, J. (1985). 'Reading Rotated Words', Journal of Experimental Psychology; Human Perception and Performance, 11 (4), pp. 490-508.

[8] Kruger, R., Carpendale, S., Scott, S. and Greenberg, S. (2003). 'How people use orientation on tables: comprehension, coordination and communication', Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, pp. 369-378.

[9] Kruger, R., Carpendale, S., Scott, S., Tang, A. (2005). 'Fluid integration of rotation and translation', Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, pp. 601-610.

[10] Mitchell, G. D. (2003). 'Orientation on Tabletop Displays', M.Sc. Thesis, Simon Fraser University, Burnaby, BC.

[11] Shen, C., Vernier, F., Forlines, C., Ringerl, M. (2004). 'DiamondSpin: an extensible toolkit for around-the-table interaction', Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 167-174.

[12] Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P. and Steinmetz, R. (1999). 'i-Land: an interactive landscape for creativity and innovation', In Proceedings of CHI '99, ACM Press, pp. 120-127.

[13] Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N. and Steinmetz, R. (2001). 'ConnecTables: dynamic coupling of displays for the flexible creation of shared workspaces', Proceedings of UIST '01, pp. 11-20.

[14] Tang, J. (1991). 'Findings from observational studies of collaborative work', International Journal of Man-Machine Studies, 34 (2), pp. 143-160.

[15] Tinker, M. (1972). 'Effect of angular alignment upon readability of print', Journal of Educational Psychology, 47, pp. 358-363.

[16] Vernier, F., Lesh, N.B., Shen, C. (2002). 'Visualization Techniques for Circular Tabletop Interfaces', ACM Advanced Visual Interfaces (AVI).

[17] Wang, Y., MacKenzie, C. L., Summers, V. A., and Booth, K. S. (1998). 'The structure of object transportation and orientation in human-computer interaction', In Proceedings of ACM CHI'98 Conference on Human Factors in Computing Systems, pp. 312–319.

[18] Wigdor, D., Balakrishnan, R. (2005). 'Empirical Investigation into the Effect of Orientation on Text Readability in Tabletop Displays', Proc. of the 9th ECSCW. (in press).

[19] Wu, M., Balakrishnan, R. (2003). 'Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays', Proc. of UIST'03, the ACM Symposium on User Interface Software and Technology. p. 193-202.