Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays

Mike Wu, Ravin Balakrishnan
Department of Computer Science
University of Toronto
mchi | ravin @dgp.toronto.edu
www.dgp.toronto.edu

ABSTRACT
Recent advances in sensing technology have enabled a new generation of tabletop displays that can sense multiple points of input from several users simultaneously. However, apart from a few demonstration techniques [17], current user interfaces do not take advantage of this increased input bandwidth. We present a variety of multi-finger and whole hand gestural interaction techniques for these displays that leverage and extend the types of actions that people perform when interacting on real physical tabletops. Apart from gestural input techniques, we also explore interaction and visualization techniques for supporting shared spaces, awareness, and privacy. These techniques are demonstrated within a prototype room furniture layout application, called *RoomPlanner*.

Keywords: tabletop interaction, gestures, multi degree-of-freedom input, collaborative and competitive applications

INTRODUCTION
Despite the proliferation and increasing functionality of electronic communication tools, face-to-face meetings remain extremely important [12], particularly when two or more people are working on a collaborative project. Tables of various types often form the focal point around which these meetings take place, due to numerous affordances that inherently support human-human collaborative activity. For example, people working at a table can sit facing each other, and share a large workspace that allows them to cooperatively work on a task while at the same time leaving sufficient semi-personal space for individual participants. In contrast, current standard computational devices such as desktops, laptops, and handhelds cater quite well to the individual but are less conducive to group activity.

Recognizing the value of tables in collaborative activity, researchers have begun exploring the use of computationally enabled tabletops [3, 5, 17, 18, 20-24]. While early computational tables such as the Active Desk [3] provided input capabilities to only one user via a pen or single finger, recent innovations in sensing hardware such as the DiamondTouch [5], SmartSkin [17] and the DViT from Smart Technologies (www.smarttech.com) make possible the detection and tracking of multiple points of input, including complex shapes such as hand profiles, from multiple users simultaneously. These new technologies hold the promise of enabling sophisticated interaction techniques that support the compound hand gestures and manipulations commonly performed by people working with physical artifacts on real tables.

In this paper, we present *RoomPlanner*, a prototype room furniture layout application. *RoomPlanner* incorporates a variety of new interaction techniques for collaborative tabletop displays that exploit the use of multiple fingers, hand shape, and gestural input (Figure 1). We also present techniques for creating and maintaining personal display and interaction spaces within the shared collaborative space. These techniques are inspired by observations of people working collaboratively on similar tasks in the real world. In addition, we discuss initial user feedback to our system. While our techniques are prototyped within this particular application, there is no reason why they cannot be more generally applied to any collaborative tabletop application.

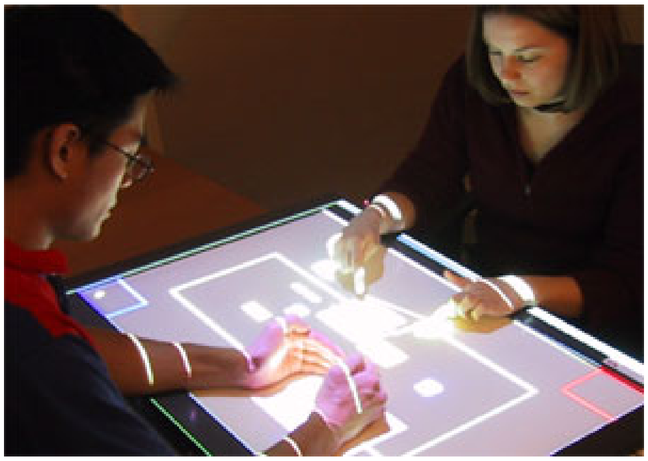

Figure 1: Two people using *RoomPlanner*.

RELATED WORK

Early research into multiple finger and hand interaction through video silhouettes was demonstrated in Krueger's VIDEOPLACE [7], in which remote users could see each other's gestures as if sitting at a table together.

The Active Desk system [3], a back projected drafting table that supported input from a single pen, was another early demonstration of tabletop interfaces. This system's form factor and limited input capabilities restricted it to single user applications such as drawing and painting programs.

Wellner's [24] digital desk prototype demonstrated the potential of integrating physical artifacts such as paper with computer vision techniques and a display projected onto a tabletop. This work focused on single user applications, but demonstrated some multi-finger techniques.

Ullmer and Ishii's [22] metaDesk prototype demonstrated tangible user interface techniques that integrated a variety of physical objects with multiple displays and projections onto tabletops. Their focus was on the physicality of the interface and not on tabletop interactions per se.

The iLand [20] and ConnecTables [21] projects describe software and hardware (both computational and furniture) infrastructure for supporting interaction on a variety of display types including tables, desks, and walls.

Vernier et al. [23] describe a variety of visualization and interaction techniques for tabletop displays. Their work is particularly interesting in that it focuses on circular tabletops – a departure from the more common rectangular tabletops investigated by others. From an input perspective, their work relied on single finger interactions.

Shen et al. [18] discuss their experiences with a "personal digital historian" application, built upon their earlier work on circular tabletops [23], that allowed users to explore archives of digital material like video clips, photographs, and documents on a tabletop display.

Three recent hardware platforms are of particular interest:

The DiamondTouch table [5] supports multiple points of input from up to four users. Its two perpendicular sets of antennas generate two arrays of data that can be used to reconstruct a 2D image of signal strengths, from which user contact with the table surface is inferred. Unfortunately, as discussed in [5] and later in this paper, detecting multiple points of input from a single user can sometimes be ambiguous. However, the system can uniquely identify each user by electrically coupling users to the table.

SmartSkin [17] is another technology that can enable tabletop sensing. Unlike the DiamondTouch, it does not distinguish input from different users. However, it offers a more complete 2D image of surface contacts, resulting in unambiguous detection of multiple input points and shapes. This work also describes several interaction techniques using hand shape and multi-finger input.

The DViT sensing technology from Smart Technologies uses computer vision techniques to sense touches, which offers advantages such as more accurately detecting a hovering finger than either DiamondTouch or SmartSkin. While DViT is not designed for tabletops per se, there is no reason why it could not be used for tabletop interaction.

RoomPlanner APPLICATION

The inspiration for this project came from our experiences in designing the furniture layout for our offices and laboratory when we moved to a new building last year. Despite all the high-end computing technology at our disposal in our human-computer interaction and computer graphics lab, we inevitably resorted to working out our designs on a conference room table with little paper cutouts of various furniture pieces. Individuals experimented with various layouts in small areas of the table, bringing versions to the middle of the table when they wanted to integrate their ideas with the larger common layout. Negotiations between people took place; pieces were aggregated into larger designs, moved around, removed, and replaced. These tasks were often performed via a variety of compound hand actions such as pushing things together or aside, grabbing a collection of items with a sweep of the hand, and tossing things to people across the table. There was also an inevitable division of the tabletop space into shared common areas and smaller private areas. Of particular interest was the observation that although everyone was working on what was ostensibly a collaborative application – that of designing the lab layout – every participant also exhibited some aspects of *competitive* behaviour. For example, when a new aspect of the layout was being discussed, people would try to get their preferred version of this sublayout into the main common layout before others got to it. Also, some people would work on their personal designs while trying to shield their work from prying eyes until they were ready to present their work to the group.

Based on this experience, we felt that a room furniture layout application would be an ideal vehicle for exploring interaction techniques that would take advantage of the multi-point and hand shape input provided by the new hardware implementations described earlier. We note that others [15, 16] have also found furniture and other layout applications as useful vehicles for related "beyond the desktop" interaction environments.

System Hardware

We used the DiamondTouch [5] table from Mitsubishi Electric Research Laboratory as the primary hardware platform for this project. The touch surface is 31" by 19", for an 8:5 aspect ratio. An NEC LT158 digital projector casts a top-down 1028x768 pixel image on the table's surface. A single graphics workstation drives the projector and processes the multiple channels of input from the table. While we prototype our techniques on the DiamondTouch hardware and further discuss ideas that are realizable under this platform, one technique we propose will only work with the more complete 2D data provided by SmartSkin. We include this technique for the sake of completeness.

Overall Functionality

We now describe the overall high-level features of *RoomPlanner*. Later sections will discuss specific interaction techniques and how they relate to these high-level application features.

RoomPlanner is designed for two people who sit across a table from each other (Figure 1). We restricted the current design to only two users mainly because of the small size of the DiamondTouch table. Users see the room from an overhead 2D orthographic view and furniture pieces are shown as outlines reflecting their relative size at the given scale. This view is identical to those used in the industry.

Both users share a public space in the centre of the table where the room walls are shown. This set up resembles that of Single Display Groupware [19]. Users have a private space located immediately in front of them. Alternative arrangements of the space could include the more continuous ones demonstrated in the DiamondSpin [23] and Personal Digital Historian [18] projects done at MERL and also the idea of having resizable personal space that can be either customized by the user or automatically determined by the system based on the intent of actions.

Since contention of ideas could occur in the public area, the private space was designed to be more of a personal working space for temporary ideas. Users can manipulate objects in the public area and in their own personal space, but they cannot control objects that are in the personal spaces of other users.

The *editing plane* is a feature which supports private furniture layout manipulation that can be extended into the public space. Editing planes are semi-transparent trays on which furniture can be manipulated. Furniture pieces can be dragged on and off of such planes. As a plane is moved about, any furniture sitting on the plane moves along with it. This functionality enables planners to copy a portion of a room to edit in their personal workspace. Dragging the plane and overlaying it on appropriate areas of the room allows for previewing of changes and new ideas.

Through a context-sensitive menu, an editing plane can be reintegrated into the room by the *add* option. An *overwrite* option replaces the furniture sitting underneath the editing plane with those pieces that lie on top of the plane. The *undo* option of the menu clears the items on the plane and removes the plane. Editing planes can also be saved as *plan objects*, which are represented by to-scale iconic outlines that can be moved and manipulated in similar ways to the furniture pieces. Plan objects are essentially abstract representations of the editing planes.

INTERACTION TECHNIQUES

From an input perspective, the interaction techniques used in *RoomPlanner* fall into four categories: single finger input techniques, two finger input techniques, single hand techniques where the shape of the user's hand is sensed, and two-handed techniques where the shape of both hands are used. Figure 2 summarizes this gesture set.

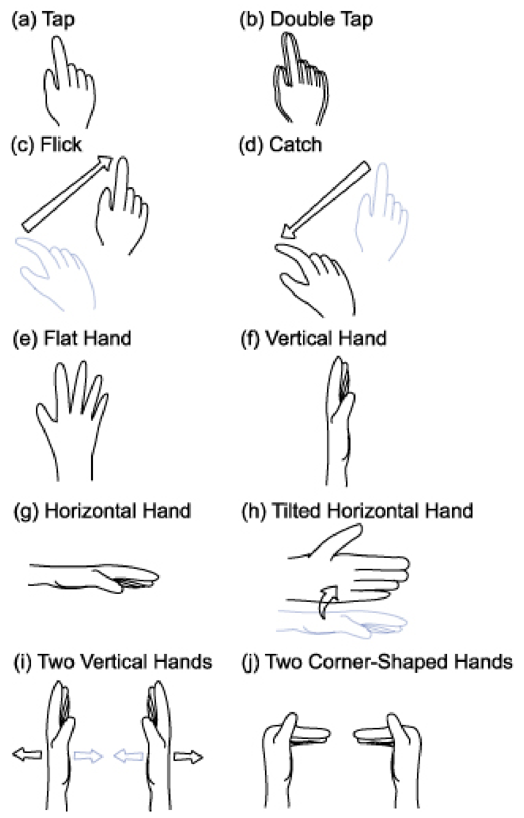

Figure 2: Gesture set. (a) Tap: single point touch. (b) Double Tap: single point touch-release-touch. (c) Flick: quickly slide single point away from self. (d) Catch: quickly slide single point toward self. (e) Flat Hand: lay hand flat on surface. (f) Vertical Hand: side of upright hand touches surface in a vertical manner. (g) Horizontal Hand: side of upright hand touches surface in a horizontal manner. (h) Tilted Horizontal Hand: tilt top of horizontal hand away from self. (i) Two Vertical Hands: symmetrically slide two vertical hands together or apart. (j) Two Corner-Shaped Hands: each hand makes a corner.

Single Finger Techniques

Tapping and Dragging

Touching a finger on an object selects it. Dragging the finger over the surface while an object is selected moves the object. If the selected object is a furniture piece, a partially transparent picture that shows what the furniture piece looks like (from a side perspective view) appears above the selected item (Figure 3).

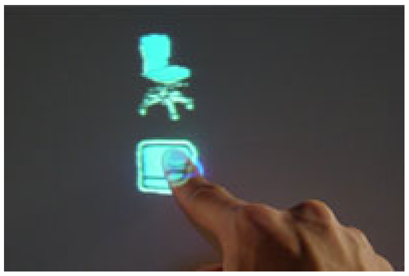

Figure 3: Partially transparent furniture preview.

Since *RoomPlanner* uses a top-down 2D orthographic view rather than a 3D perspective view, this visual feedback serves to remind users of what the furniture piece actually is, as it can be difficult to distinguish or recognize some furniture given only the overhead view of its outline.

Context-Sensitive Menus

A double tap pops up a context-sensitive pie menu [4]. The various context-sensitive menus in the *RoomPlanner* are as follows (menu items indicated in brackets): the FurnitureManipulation menu (adjust rotation parameter, delete), the EditingPlane menu (overwrite, add, undo, save plan), and the PlanObject menu (open, delete). By default, when the user double taps on an unused portion of the floor plan, the FurniturePalette menu is activated (chairs, tables, shelves, misc).

Double tapping is used over dwell time as a mechanism to activate the menu because double tapping is a more distinct gesture. The use of transient pie menus that pop-up at the location of the user's finger reduces the clutter of permanent on-screen menus and minimizes the amount of time required to make a menu section. Unlike Marking Menus [8] where menu items can be selected by simply gesturing in the appropriate direction and lifting the finger up, we require that the users actually have to move past the boundary of the menu to complete selection. This approach enables selection without requiring the finger to be lifted up to indicate completion. We use this feature advantageously in the FurniturePalette menu where after selecting the desired furniture palette, a semi-transparent toolglass [1] palette containing various furniture items immediately attaches itself to the user's finger (Figure 4).

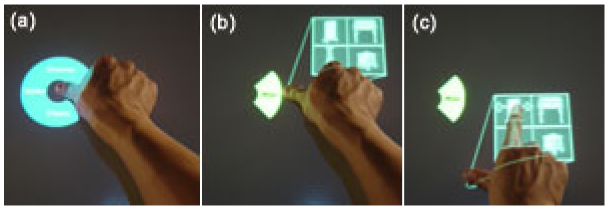

Figure 4: FurniturePalette tool. (a) A double tap on the table brings up a context-sensitive menu. (b) Sliding the finger in one of the four directions causes a corresponding toolglass to be attached to the finger. (c) A second finger is used to make a selection within the toolglass.

In a single gesture, without ever lifting up their finger, the user selects the menu item and smoothly transitions into manipulating the palette. A similar compound phrase [2] approach is used in Control Menus [14] to combine menu selection and parameter adjustment in a single action. Our two-finger parameter adjustment widget to be described shortly is similarly activated.

Flicking and Catching

Plan objects can be manipulated in a similar manner as furniture objects, except that they have an extra capability: we use a dynamic flicking gesture to send a plan object to another user seated around the table. This flicking gesture is analogous to the throw action that is commonly performed by people passing objects around when working at regular physical tabletops. When the user moves a plan object toward the other user past a certain speed threshold, the plan object is thrown to the private space of the other user. The plan object then sits with half of it being displayed on the table and half of it being displayed on the raised frame surrounding the table. This is possible as we project an image with a 4:3 aspect ratio onto an 8:5 DiamondTouch viewing surface – thus offering us the ability to project items above and below the touch-sensitive boundaries of the DiamondTouch. Using this frame display area enables the insertion of material into another user's personal space as unobtrusively as possible while at the same time providing awareness of the action. The disadvantage, of course, is that touch is not detected on the frame of the device and so users cannot directly manipulate objects displayed on it. Our partial use of the frame area (Figure 5) is a reasonable compromise solution.

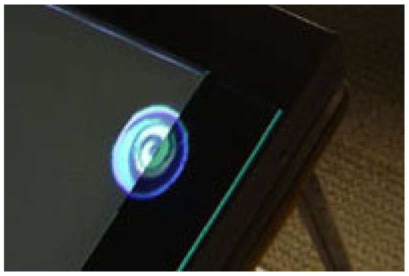

Figure 5: After flicking a plan object, half of it sits on the touch-sensitive surface and the other half sits on the frame of the device.

The opposite gesture is the catching motion. Here, the user touches the surface with their finger and draws a straight line in a direction toward him/herself. If the speed of the movement surpasses a set threshold, the plan object sitting across the table that intersects the line is copied. The copy of this object lands in the private space of the user performing the catch gesture. It should be noted that there is storage space located at the corner left of each user (but still within their private space). Plan objects inhabiting these areas cannot be copied by the other user. This reinforces the idea of object ownership and thus privacy.

Instead of copying the object, an alternative to our implementation is to actually retrieve the object from the other user's private space. In this case, the user who has the plan object sitting in their space could authorize the retrieval somehow, perhaps by simply touching the object.

In terms of passing objects on the table, we had also considered one other method. One of DiamondSpin's [23] document visualization techniques is a "black hole" fisheye technique. To build upon this metaphor, a "wormhole" layout may be an alternative to the flick/catch gestures in that it could support object sharing through connected endpoints of the "wormholes". Each endpoint would be located in different user spaces and objects may be shared by dragging them to the mouth of a wormhole. This parallels a scrollbar widget demonstrated in [11]. We did not incorporate such a system because the RoomPlanner did not make use of the "black hole" layout.

Given the smallish size of our DiamondTouch table, one could argue that these gestures are not required since users could simply reach to the other end of the table and place or grab what they need. However, we developed these flicking and catching gestures because they scale well into situations that involve more users on a larger table where users would not be able to physically reach all edges of the table while seated. In such a setup, the direction of the flicking gestures can distinguish the person to whom the information is intended. Also, the gesture could be broken into two steps so that users could ensure that they have properly directed the information before confirming the action.

Two Finger Techniques

Freeform Rotation and Scaling

To achieve a free rotation of a furniture piece, two fingers (not necessarily from the same hand) are used. The first finger determines the centre of rotation, while the second determines the rotation angle. While an object is selected, touching a second finger onto the tabletop initiates the rotation. The change in rotation angle for the object is calculated by the change in angle between the two fingers, much like in Dual Touch [10]. After the freeform rotation is initiated, the original pivot finger may be lifted. Though the object cannot be moved until the first finger touches the furniture piece again, this allows a greater range of rotation as the movement of the second finger is not constrained (Figure 6).

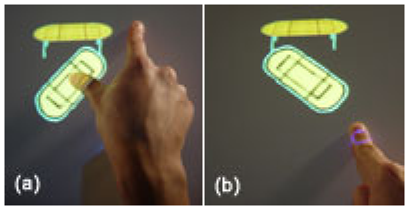

Figure 6: Freeform rotation. (a) Two fingers are used to rotate an object. (b) Though the pivot finger is lifted, the second finger can continue the rotation.

An alternative technique is to have a furniture piece "lock" onto the fingers that are touching it so that the furniture rotates and translates with the movement of those fingers as a whole. However, this approach does not work well with objects that are too small to be simultaneously touched by two fingers. This problem could be addressed by using the bounding box defined by the two fingers to enclose a piece of furniture, and then using the fingers to specify a transformation by rotating and translating this box.

We also considered another alternative technique that has a closer mapping to the real world. The rotation of a physical piece of paper can be achieved by touching the piece with one finger and rotating that finger, using friction to turn the paper. Such a technique would require the system to detect finger orientation. One way to do so is to look at where the non-touching fingers are hovering over the surface in relation to the touch point.

Though not built into RoomPlanner, scaling may also be specified by two fingers in that the scale factor can be determined by the distance between the two fingers, in a manner similar to that proposed by Rekimoto [17].

Tool Palette Manipulation and Selection

As described earlier, in a single gesture a user can smoothly select a furniture toolglass palette from the FuniturePalette menu and have the palette remain attached to the finger that made the selection. The palette can then be moved around the display to the desired location for placing furniture. By using a second finger the user can place furniture at the desired location by pressing on the appropriate palette item (Figure 4c).

We argue that this two-fingered approach to toolglass palette manipulation is an improvement on the two-*handed* approach originally proposed by Bier et al. [1] and used in the T3 prototype of Kurtenbach et al. [9] in that the two fingers are more closely linked in the user's kinematic motor control chain [6] than the two hands are. Our system does however provide the flexibility of achieving the same interaction with fingers of different hands if desired.

This two-fingered palette manipulation also allows for multiple placements of furniture around the room without having to lift up the initial finger or requiring the system to enter a mode. In essence the first finger does double duty as a locator for the toolglass palette and also kinesthetically holds the palette in a transient or quasimode.

Parameter Adjustment Widget

As an alternative way to rotate an object, the user can use a parameter adjustment widget that takes advantage of inherent physical finger properties to dial a parameter. As with the toolglass palette, this widget is accessed through a context-sensitive menu. When activated, the widget is attached to and follows the finger that made the selection.

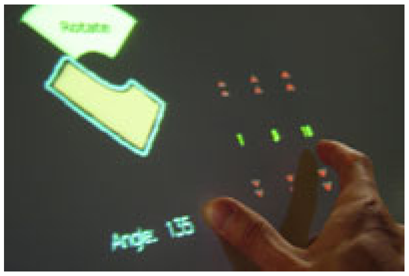

Figure 7: This parameter adjustment widget allows two- fingered manipulation.

As shown in Figure 7, the widget displays six arrows, arranged in three groups. Each group consists of two arrows that point in opposite directions. The arrows closer to the first finger are smaller than those farther away. When the user's second finger touches one of the arrows, the parameter is either increased or decreased by some amount.

The distance between the two fingers determines the granularity of adjustment. In other words, the closer the fingers are to one another, the smaller the change in angle. The three sets of arrows displayed, each of different size, serve as a visual indicator of this property.

Single Hand Techniques

Flat Hand

A user can temporarily rotate the room layout by placing a hand flat on the table and translating that hand. As the line between the centre of the hand and the centre of the room rotates, the public space pivots in the middle and turns with it. Once the flat hand is placed on the table, the planner need not keep their hand flat to continue the rotation, as long as they continue touching the surface (i.e., the flat hand serves mainly as a quick way to choose the rotation action). This allows the user to adopt a more comfortable hand posture – similar to the case of the freeform rotation of objects when the pivot finger is lifted. When the user removes contact from the surface, the room springs back into its original orientation.

As an alternative method for room layout rotation, users could turn their hand in place on the surface, using the hand direction to indicate rotation angle. This was not done because it did not map well to real world situations where large blueprints are physically manipulated by sliding the hand about. Another possible technique is to not have the room pivot at the centre but instead have it freely arranged by the orientation and location of the hand, much like grabbing a large blueprint and re-orientating it by moving and turning the hand. Though hand orientation is difficult to accurately detect using DiamondTouch, it is possible with SmartSkin.

Our observations of people interacting on the table with our application indicated that they respected each other's space when working in the common area. Thus we did not implement any technological solution to deal with contention or conflict resolution, leaving it up to the users to resolve via usual social norms. In our current implementation, when both planners try to rotate the room at the same time, *RoomPlanner* averages the angles expressed by both planners. However, other techniques such as halting the rotation when there is contention may be viable should a technological solution be necessary.

Vertical Hand

When the side of a hand is placed on the surface of the table oriented such that the contact surface is a vertical line, the user can sweep furniture pieces. As objects make contact with the hand, they are pushed aside and are swept at the same pace as the movement of the hand (Figure 8). This parallels the real world action of pushing physical material on a regular tabletop. This gesture can be independently applied to the private and public spaces so that when a planner sweeps furniture in their private space, objects in the public area are not moved. The user specifies to which space the intended action is to be applied by simply performing the gesture within that space.

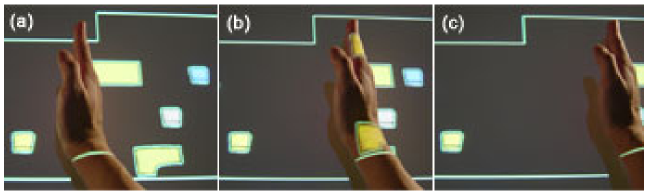

Figure 8: Vertical hand sweeping. (a) Initial position. (b) When the hand makes contact with furniture, the pieces move with it. (c) Final position after sweeping.

Horizontal Hand

The side of the hand can be placed on the table such that the contact surface forms a horizontal line. This gesture creates a rectangular box below the hand through which objects within the box display their properties (Figure 9); this is similar to the idea of Magic Lens [1]. As the horizontal hand moves, the box follows below it.

It is interesting to note that when a user performs this gesture, the hand acts as a barrier that blocks others from seeing the displayed information. Although others could view any information projected on the tabletop if they tried hard enough, the horizontal hand gesture provides some amount of privacy on the tabletop. To violate this privacy, another user would have to stand up and look over the first user's horizontally held hand, which would be a social faux pas. We imagine that this gesture may also be useful as a means of accessing a set of menus or to cast private votes. By incorporating both voting and regular menu selections through this same gesture, other users may not recognize hidden negotiations since they would appear as regular menu actions. This is an interesting issue to explore in the future as it points towards the notion of supporting competitive behaviour within an overall collaborative application framework.

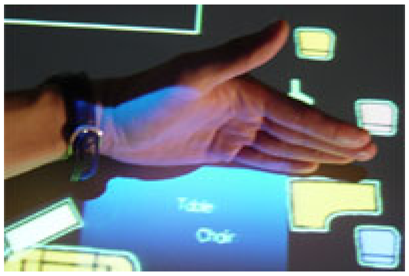

Figure 9: The horizontal hand physically blocks others from seeing the box of furniture properties below it.

Tilted Horizontal Hand

The top-down projection setup enables advantages that would not be possible if the display were back projected. For example, when a piece of paper is placed on the table, objects are projected on top rather than being occluded. Similarly, when a user's hand occludes the surface, it can act as display area.

When the hand is in the horizontal position, tilting the top of the hand away from the user is a gesture that allows the system to project private information onto the hand. As this information can only be seen by the user, the hand is used to create physical space upon which private information can be projected (Figure 10). Because this gesture does not require much movement, it may be more appropriate than others in supporting private information in a more subtle and discreet way.

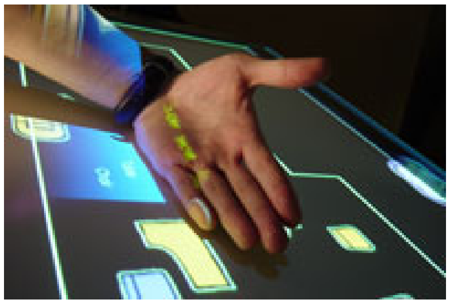

Figure 10: The tilted horizontal hand gesture uses the hand as physical space upon which to project information.

We imagine that tilting the top of the hand back toward the user can be used to present information for others to see. This feature was not implemented in our prototype because the table was small enough so that planners could quite easily see each other's private work areas and any information that may be presented therein. A large table could, however, benefit from this gesture.

The tilting recognition is made possible not through evaluation of contact surface, but by evaluation of the elements of the antenna array above or below the horizontal hand; if enough of those elements are excited beyond a specified threshold, the system performs the associated command.

Two-Handed Techniques
We implemented two interaction techniques employing symmetric two-handed gestures. We considered, but did not implement, asymmetric gestures in which one hand provides the frame-of-reference while the other hand specifies an action within this context. For example, laying a fist on the table while flicking a plan object with the other hand may mean that the plan object is thrown to another user, but a copy is kept in the user's private space.

Two Vertical Hands
Sweeping two vertical hands away from each other will move all the furniture pieces in a manner similar to raisins in bread dough when the bread expands as it bakes. All the furniture moves at a constant velocity, but if a hand makes contact with a furniture piece, that piece is carried along with the hand. This gesture is quite similar to the single vertical hand gesture and so we tried to maintain consistency in action and behaviour. Both these gestures (as well as the collapsing gesture described below) can be independently applied to the public and private spaces.

The complement to the expanding gesture is the collapsing gesture in which two vertical hands are brought toward one another. This gesture collapses the space by horizontally displacing furniture pieces so that they are clumped together at the centre, as shown in Figure 11. One could also imagine vertical displacement and clumping, but that would involve the user either having to turn their body sideways or move their hands in an awkward manner.

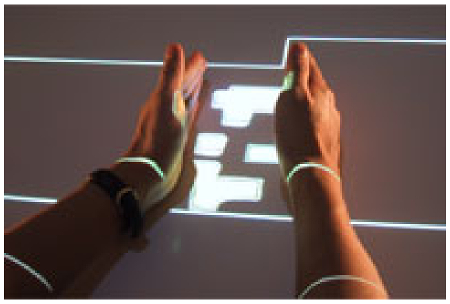

Figure 11: Sweeping two vertical hands together collects furniture pieces in the centre.

One idea that was not built into the current prototype is to support gestures where the two hands move in the same direction, for example both to the right, carrying along whatever furniture is between them.

Two Corner-Shaped Hands
Two corner-shaped hands can be used to create a rectangular editing plane (Figure 12), which copies a portion of the shared room layout in order for users to individually work on sections of the room (possibly even the same section).

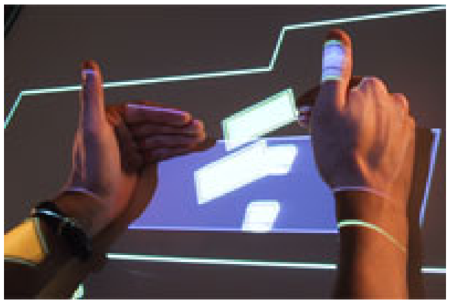

Figure 12: Two corner-shaped hands are used to define and move an editing plane.

The initial dimensions of the editing plane are defined as soon as the corner-shaped hands touch the surface. The horizontal distance between the hands defines the initial width of the editing region, and the length of the palm defines its height. The width of the box can be modified by moving the hands horizontally apart, but currently no method is in place to allow for modifying its height. All furniture pieces located within this region are copied onto the editing plane, in the same orientations and relative locations as the originals.

Dragging the two corner-shaped hands over the surface moves the editing plane (the other technique of moving the plane is to select and drag it using one finger). The plane can be pulled into the user's private space for editing. This gesture can also be used to push the editing plane back into the shared viewing space for previewing or reintegration.

USER FEEDBACK

We sought to obtain some user feedback both to inform the design of *RoomPlanner* and to obtain reactions to the gesture set. Since our intention in doing this work was to explore multi-finger and whole hand interaction techniques, rather than furniture layout applications per se, we chose novice users from which to gather feedback so that we could gauge how difficult it was for someone to learn and operate our gestural interface.

Five people participated in hour long trial sessions each. In each session, a participant acted as one planner, and a researcher (the first author of this paper) acted as the other planner. This "constructive interaction" technique, commonly practiced in usability evaluations, was chosen over having two novices use the system simultaneously while the researcher observed as it allowed a more interactive session where the researcher could prompt a quiet user by asking them to comment on, for example, a portion of a room of which they were editing. As both planners shared the same high-level goals, the internal prompting was less disruptive than having an external observer cut into the planner's discussion.

At the beginning of the session, participants were informed that the research was exploratory in nature and not designed to solve the demands of a real working architectural situation. After being introduced to the application, they were then shown each of the interaction techniques while also receiving details of how the gestures were specifically related to the application.

The participants were then given a scenario that described some room that needed to be arranged with furniture. Scenarios varied in the number of desks and workers that needed to fit within the space; also, some rooms were more densely packed while others involved concerns such as noise levels. The room type was different as well, some rooms being lab spaces and others being classrooms. This allowed us to examine the interaction techniques under somewhat varying contexts.

We observed that the participants required little practice to learn the gesture set and were able to use the gestures effectively to create room layouts. User feedback was quite consistent among the five participants, with the following emerging as the key issues:

Occlusion Issues

When operating the pie-menus, the tapping hand sometimes occluded some of the displayed menu items. One user suggested that instead of arranging the menu items in a complete circle around the point of tapping, the items could be displayed in a semi-circle fashion *above* the tapping finger. This is an interesting solution in that it does not require knowledge of which hand made the tap in order to avoid occlusion, as would be required if the items were instead displayed to the left or right of the tap position. However, knowing which hand initiates a gesture may facilitate other functionality. For example, different menus could be assigned to each hand.

Size, Scale and Rotation Control

Users expressed a desire for additional functionality for the editing planes, including the ability to rotate and resize them after creation. Two users suggested touching the corners of the editing plane with four fingers (a thumb and index finger of each hand) and then moving the fingers to resize the plane. It was interesting to observe this symmetric motion despite the fact that only two corner points need to be manipulated in order to achieve the same effect. Some users also expressed a desire for snapping to primary axes while rotating objects.

Catching Gesture

One user described a "fishing line" mental model for how the catching gesture worked. This user requested for some visual feedback that showed which plan object was actually copied, as it was noted that several plan objects could be clustered close together. Also, catching a plan that was already copied should provide visual feedback, perhaps by having the computer highlight the copy.

Parameter Adjustment Widget

When activating and manipulating the two-fingered widget, it was quite easy to double tap with a thumb and have the free fingers of that same hand manipulate the widget. We observed that if the index finger did the tapping instead of the thumb, it was more difficult to modify the parameter using fingers of the same hand. Another approach we may consider when the user taps with their index finger is having the middle finger specify the magnitude factor and the thumb specify direction (increase or decrease of a value). This would be a three-fingered widget. As the system hardware cannot determine which finger is associated with which touch point, some assumptions about how fingers are arranged would be necessary.

Single Hand Techniques

Two users reported wanting to pin the floor plan down after rotating it with the flat hand gesture. All users wanted the ability to pull objects down or push objects away using the horizontal hand to help organize pieces of furniture; in other words, gestures that were just like those of the vertical hand. Adding this functionality would require a way to multiplex with the current functions assigned to the horizontal hand. Dwell time may be a viable solution: the horizontal hand would by default act as a way to move furniture along the y-axis, but if it paused for a short period, the existing functions of displaying object properties and private information when the hand is tilted would be activated. This has the interesting side-effect of enabling users to hide their intent since others around the table would not easily know if a horizontal hand gesture is being used to move furniture or to do private tasks.

Two Corner-Shaped Hands

We observed that the corner-shaped hands did not provide much control when the objects were quite closely packed. One user suggested a more asymmetric gesture that would involve one corner-shaped hand and a single finger that points to the opposite corner of the editing plane.

DISCUSSION and FUTURE WORK

While our work has explored a variety of pertinent issues to be considered when designing interfaces for multi-user tabletops, much research clearly remains to be done before tabletop applications can be reasonably utilized in the real world. Some interesting areas to be explored are further support for awareness, extension to larger sized tables, and multi-person collaborative gestural interactions. At the end of this section, we also discuss sensing algorithms and how the hardware constraints informed design of the interaction.

Awareness and Privacy

When working around a table, people are often aware, at least at a course level of granularity, of the actions being performed by others at the table. One can see another person writing notes, or using liquid paper before making a correction. However, one usually cannot distinguish what it is that someone else is writing, or what it is that they decide to erase. *RoomPlanner* provides some support for such "selective" awareness in the form of the horizontal hand technique. We can imagine providing further support by allowing users to define a set of private gestures that they would then associate with particular functionality. This private gesture set would coexist with the common public gesture set. While everyone can observe a user performing a particular private gesture, that gesture would have no meaning to anyone but that user. One could further extend this concept to having sets of gestures for particular groups of people, thus supporting competitive actions by groups working on a globally cooperative application.

Extension to Larger Tables

Our interaction techniques are easily extensible to larger tables such as conference tables with more people. However, larger tables have some key differences. A more obvious difference is that the increased separation between users will make it more difficult for participants to see each other's private spaces, perhaps with the exception of neighbours. In this case, actual physical dividers may be set in place that can be raised to block the sight of people sitting next to one another, or the participants may be arranged in some order that allows partial sharing of information among neighbours, but not by the entire group. As mentioned before, the flick/catch gestures would be more prominent in situations that involved a larger table.

In our current implementation of *RoomPlanner*, we did not provide any explicit mechanism for resolving contention, leaving that up to human-human communication to resolve. At a larger table, however, more explicit techniques may be required. For example, if two people want to manipulate a common object at the same time, we could implement a solution where users control a proxy node, similar to the Voodoo Dolls approach of Pierce et al. [13]

Multi-Person Gestures

We mentioned the idea of having one hand specify the frame-of-reference for another hand's action. This may be extended to gestures based on multiple people: one person might specify the context of another person's gesture. An example of such a gesture could be when a user would like to share a set of objects by placing an open palm onto the tabletop in the public space. Other users may then use a flicking gesture to add objects to that set or a catching gesture to make copies of those objects.

Sensing Algorithms

As mentioned before, the DiamondTouch system provides only two arrays of sensor data, one for each antenna axis. This has several design implications.

In the 2D image reconstructed from these two sensor arrays, there may exist ambiguities in the 2D data. For example, when two fingers are touching the surface, two peaks are seen in the horizontal antenna array and two peaks are seen in the vertical, but it may not be clear which peaks of the horizontal array correspond to which peaks of the vertical. To disambiguate between these points, we assume that no two fingers contact the surface at exactly the same time. We thus figure out which peaks correspond to each other by careful temporal examination when new peaks appear.

The location of touch points are tracked by looking at each new image of data and comparing it to the previous image, predicting where touch points will likely be based on their speed and direction. This information is used to appropriately calculate the location of the touches in the newer image should there be any uncertainty.

Our gestures were most easily recognized by the system when performed along the rectilinear array of the DiamondTouch. This is an intrinsic property arising from the arrangement of the antenna arrays at the hardware level.

Another issue in multi-finger and shape-based gesture recognition arises from the fact that every person has unique hand and finger dimensions and characteristics. Thus, customizable gesture scales (or possibly even customizable gesture shapes) could be valuable. This suggests the need for a calibration step or learning on the part of the system in order to determine the maximum, minimum, and preferred ranges of hand properties for particular users.

CONCLUSION

Our work has explored a variety of interactions for multi-user tabletop displays that leverage and extend the types of multi-finger and whole hand actions people perform when interacting with physical entities on real tables. An interesting aspect of our work is the exploration and support of not only collaborative actions, but also competitive ones. In addition, our techniques support the notion of public and private spaces, and awareness of the actions of other users. Initial user feedback suggests that though learning the techniques does not require much time, users desired additional functionality in the room planning application. The interaction techniques presented are also relevant to a broader range of tabletop applications that do not necessarily involve room layout planning, such as collaborative editing systems or group negotiations.

ACKNOWLEDGEMENTS

We thank Joe Marks and Kathy Ryall at Mitsubishi Electric Research Laboratories for providing DiamondTouch, Tovi Grossman and Anastasia Bezerianos for helpful suggestions and video production, Davide Bolchini for his inspirational background music interleaved with the video, Michael McGuffin for editorial help, and members of the Dynamic Graphics Project laboratory (www.dgp.toronto.edu) at the University of Toronto for valuable discussions.

VIDEOS

A digital video clip demonstrating this research can be downloaded from www.dgp.toronto.edu/research/tabletop

REFERENCES

1. Bier, E., Stone, M., Pier, K., Buxton, W., & DeRose, T. (1993). Toolglass and Magic Lenses: The see-through interface. *ACM SIGGRAPH Conference*. p. 73-80.
2. Buxton, W., Chuncking and phrasing and the design of human-computer dialogues, in *Readings in human-computer interaction: Towards the year 2000*, R. Baecker, *et al.*, Editors. 1986, Morgan Kaufmann: San Fransisco, CA. p. 494-499.
3. Buxton, W., Living in augmented reality: Ubiquitous Media and Reactive Environments, in *Video Mediated Communication*, K. Finn, Sellen, A., Wilber, S., Editor. 1997, Erlbaum: Hillsdale, NJ. p. 363-384.
4. Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). A comparative analysis of pie menu performance. *ACM CHI Conference*.
5. Dietz, P., & Leigh, D. (2001). DiamondTouch: a multi-user touch technology. *ACM UIST Symposium on User Interface Software and Technology*. p. 219-226.
6. Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior, 19*(4). p. 486-517.
7. Krueger, M., VIDEOPLACE and the interface of the future, in *The art of human computer interface design*, B. Laurel, Editor. 1991, Addison Wesley: Menlo Park, CA. p. 417-422.
8. Kurtenbach, G., & Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. *ACM CHI Conference.* p. 35-42.
9. Kurtenbach, G., Fitzmaurice, G., Baudel, T., & Buxton, W. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *ACM CHI Conference*. p. 35-42.
10. Matsushita, N., Ayatsuka, Y., & Rekimoto, J. (2000). Dual touch: A two-handed interface for pen-based PDAs. *ACM UIST Symposium on User Interface Software and Technology*. p. 211-212.
11. Myers, B. (1990). All the widgets. *ACM SIGRAPH Video Review , CHI'90 Special Issue, 57*.
12. Olsen, G., & Olsen, J. (2000). Distance matters. *Human-Computer Interaction, 15*(2&3). p. 139-178.
13. Pierce, J., Stearns, B., & Pausch, R. (1999). Two handed manipulation of voodoo dolls in virtual environments. *ACM Symposium on Interactive 3D Graphics*. p. 141-145.
14. Pook, S., Lecolinet, E., Vaysseix, G., & Barillot, E. (2000). Control menus: Execution and control in a single interactor. *ACM CHI Conference, Extended Abstracts*. p. 263-264.
15. Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhardt, U., & Meier, M. (1998). BUILD-IT: A planning tool for construction and design. *ACM CHI Conference, Extended Abstracts*. p. 177-178.
16. Rekimoto, J. (2000). Multiple-computer interfaces: "Beyond the desktop" direct manipulation environments. *ACM CHI Conference, Extended Abstracts*. p. 6-7.
17. Rekimoto, J. (2002). SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *ACM CHI Conference*. p. 113-120.
18. Shen, C., Lesh, N., Vernier, F., Forlines, C., & Frost, J. (2002). A social sense of time: Sharing and building digital group histories. *ACM CSCW Conference on Computer Supported Cooperative Work*. p. 324-333.
19. Stewart, J., Bederson, B., & Druin, A. (1999). Single display groupware: A model for co-present collaboration. *ACM CHI Conference*. p. 286-293.
20. Streitz, N., Geißler, J., Holmer, T., Konomi, S.i., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., & Steinmetz, R. (1999). i-LAND: an interactive landscape for creativity and innovation. *ACM CHI Conference*. p. 120-127.
21. Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., & Steinmetz, R. (2001). Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. *ACM UIST Symposium on User Interface Software and Technology*. p. 11-20.
22. Ullmer, B., & Ishii, H. (1997). The metaDESK: Models and prototypes for tangible user interfaces. *ACM UIST Symposium on User Inteface Software and Technology*. p. 223-232.
23. Vernier, F., Lesh, N., & Shen, C. (2002). Visualization techniques for circular tabletop interfaces. *Advanced Visual Interfaces*.
24. Wellner, P. (1993). Interacting with paper on the digital desk. *Communications of the ACM, 36*(7). p. 87-96.