Nailing Down Multi-Touch: Anchored Above the Surface Interaction for 3D Modeling and Navigation

Bret Jackson* David Schroeder† Daniel F. Keefe‡

Department of Computer Science and Engineering, University of Minnesota

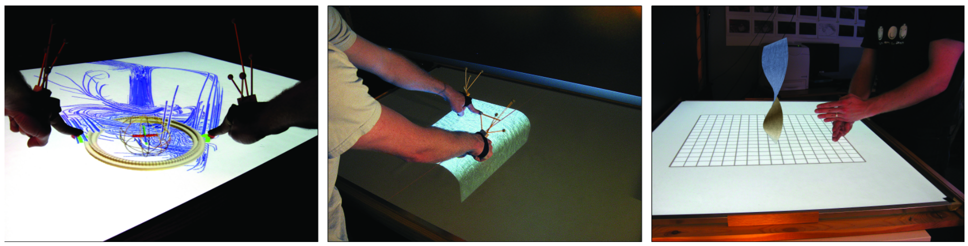

Figure 1: Anchored multi-touch gestures for 3D head-tracked stereoscopic applications. Left: Pivoting the fingers above the surface rolls a 3D visualization. Center: A user bends a mesh while simultaneously specifying the curvature of the bend. Right: A user twists a 3D mesh by placing his hands on top of each other and rotating. (A digital rendering is superimposed to illustrate the stereoscopic effect of the display.)

Abstract

We present *anchored multi-touch*, a technique for extending multi-touch interfaces by using gestures based on both multi-touch surface input *and* 3D movement of the hand(s) above the surface. These interactions have nearly the same potential for rich, expressive input as do freehand 3D interactions while also having an advantage that the passive haptic feedback provided by the surface makes them easier to control. In addition, anchored multi-touch is particularly well suited for working with 3D content on stereoscopic displays. This paper contributes two example applications: (1) an interface for navigating 3D datasets, and (2) a surface bending interface for freeform 3D modeling. Two methods for sensing the gestures are introduced, one employing a depth camera.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies

1 Introduction

From the rock climber gripping a hold to the musician plucking strings, our hands are extremely versatile, allowing us to interact with the world in amazing and expressive ways. Multi-touch interfaces are now beginning to take advantage of this ability and, in doing so, revolutionize the way we interact with computers, enabling much richer and expressive input than with traditional mice and keyboards. In the future, we expect the higher-bandwidth input, somesthetic information, and collaboration support that multi-touch provides to continue to drive adoption and to open up new applications for this style of expressive human-computer interaction. To facilitate this expansion, our work seeks to understand how multi-touch input might be usefully complemented by 3D input from the hands above the touch surface.

Our work is motivated by 3D applications, such as scientific visualizations and 3D modeling. Traditional multi-touch interactions are less successful in these applications, in part because the mapping from 2D surface input to 3D action is less clear than in the purely 2D case. This problem is compounded in head-tracked stereoscopic environments. In stereoscopic environments 3D objects viewed on a multi-touch table can be made to appear as though they float above the table or under the table. Even the traditional translate-rotate-scale gestures used so often in 2D multi-touch environments break down in this situation. Scale, for example, might be interpreted either as scaling the 3D scene or as translating the 3D scene up and out of the table toward the viewer's eyes. Extending multi-touch interfaces to include above-the-surface 3D inputs may provide the needed increased richness and expressiveness in order to address situations such as these.

Although some work has studied free-hand interactions in the air above a multi-touch surface (discussed in Section 2.1), in this paper we explore the interactions space *just* above the surface, where the shape or movement of the hand is captured simultaneously with touch inputs on the surface. We call this type of interaction *anchored*, as the gesture is performed while maintaining an anchoring touch to the surface. This allows us to leverage the benefits of using the surface as a passive haptic support, potentially improving stability and control as well as lowering fatigue when compared to freehand 3D input in the air [9]. In the applications we demonstrate, users move fluidly between traditional 2D gestures on the surface, and anchored 3D gestures above the surface.

We present two applications of anchored multi-touch: (1) an interface for navigating 3D datasets, and (2) a surface bending interface for freeform 3D modeling. Both of these applications, shown in Figure 1, use anchored gestures to interact with 3D content on a head-tracked stereoscopic multi-touch display. In addition to these applications our contributions include an algorithm for sensing anchored gestures above a touch surface using depth-sensing cameras. The remainder of this paper begins with a discussion of related work. We then present the anchored gesture applications. Finally, we conclude with a discussion of lessons learned and future research directions.

*e-mail:bjackson@cs.umn.edu

†email:daschroe@cs.umn.edu

‡e-mail:keefe@cs.umn.edu

2 RELATED WORK

2.1 Interaction Above the Surface

Others (e.g. [3, 7, 12]) have looked at expanding the expressiveness of multi-touch, particularly for 3D interaction by tracking gestures in the air above the surface. Our approach has nearly the same potential for rich expressive interaction as these free-hand gestures. However, motivated by [9] which found that resting the hands on the surface improved control in steering tasks above the surface, our gestures are anchored with touch contacts to the surface. Additionally, these anchored touch points allow fluid movement between traditional 2D multi-touch gestures and more expressive 3D gestures, similar to Marquardt et al.'s [10] idea of the continuous interaction space. However, the transition is even more fluid with our gestures because the hands never need to leave the surface.

Recent advances in tracking technology have made these types of above-the-surface interactions possible. For example, the Z-touch [13] hardware by Takeoka et al. uses infrared laser planes to create a depth map of objects near the surface of the table and can sense the posture of the hands. We believe that this type of hardware opens up new and exciting opportunities for different types of interaction such as the anchored gestures we explore in this paper. Although their main contribution was the hardware, Takeoka et al. also describe several interesting interaction techniques for drawing, controlling map zoom level, and Bézier curve control by varying the finger angle relative to the surface of the table. Like Takeoka et al., our depth camera-based tracking approach does not require users to wear additional hardware. An added advantage of our system is that recent advances in the entertainment market have made depth-sensing cameras, such as the Microsoft Kinect, available at low-cost.

2.2 Multi-Touch in 3D Space

Several others have explored the idea of how the physical relationship of the hands to the surface can enable different interface techniques. In one of the first systems in this style, Grossman et al. [5] defined several new techniques for interacting on and around the surface of a spherical display. In this work as in ours, the angle finger makes with the surface can be used as an important input to the system. Our work explores a variety of new gestures and applications for use with planar multi-touch surfaces; however, we believe that many of the interaction metaphors we describe could also extend to non-planar touch surfaces.

3 HAND TRACKING

This section introduces two methods for sensing 3D motion just above the table surface. The first method utilizes commercially available optical motion capture technologies; the second uses a depth camera and includes a discussion of our extension of an existing finger tracking algorithm to work with anchored gestures.

Both approaches are implemented on the stereoscopic display table pictured in Figure 1. The display is head-tracked, so as the user moves his head around the table, the perspective projection of the scene is updated accordingly. A 7-camera OptiTrack system (NaturalPoint) is installed in the room to support head-tracking via reflective markers.

The first method for tracking hand gestures also makes use of the Optitrack system. In this method, a constellation of reflective markers is affixed to the back of the hand using a thin elastic (see left two images in Figure 1); the system tracks these markers, reporting 6 degrees of freedom tracking data for each hand.

There are two clear limitations to this technology. First, it requires wearing extra hardware on the hands. Second, it treats each hand as a rigid body, i.e., it tracks the whole hand, not individual fingers. Both of these limitations should be able to be solved using current low-cost commercially available depth cameras, such as the Microsoft Kinect. In the future, we expect that anchored gestures will be reliably captured via this type of depth-sensing technology or perhaps a multi-layer strategy built into the frame of the multi-touch table, as in [13].

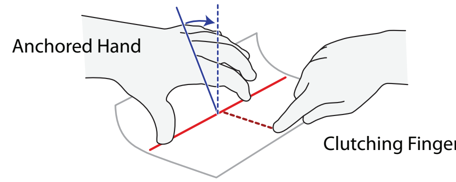

Figure 2: Surface bending by pivoting the hand around the anchored thumb and index finger. A clutch finger on the second hand locks the bend axis in place and specifies the amount of curvature.

Our work takes a first step in this direction; the surface twisting interface uses a 640×480 pixel depth image collected at 30 Hz to track anchored gestures of the hands. This particular gesture uses two hands, one placed on top of each other, as in Figure 1 right. The tracking algorithm follows a strategy inspired by a recently introduced finger tracking algorithm [6], with extensions here to identify the larger hand features and segment the two hands. First, the horizontal gradient is extracted from the depth image. Then, each row is searched for a pattern consisting of a high-magnitude gradient value followed by a smooth region at least 5 cm long. Since our camera is placed off to the side of the multi-touch table, we are able to identify the rightmost (in camera space) section matching this pattern as the tip of the hand.

With the hand tip identified, a smoothing operation is applied to the depth values, and the region of the image considered to contain the hand is grown by adding neighboring rows of pixels that exhibit similar depth patterns. Since the hands touch each other, after this step, the selected region will include both the top and bottom hands. The two can be separated by greedily growing from the top and bottom of the hands region, using row-wise difference to determine the similarity between each pair of rows. Finally, once the two hands are identified, the average horizontal gradient is calculated for each hand. The arctangent of these gradients is the rotation angle for each hand.

4 APPLICATION 1: ANCHORED 3D MODELING

The first application we explore is 3D modeling. There is a long history of research in free-form modeling tools using touch, pen, and freehand input (e.g., [8]), yet creating controllable, natural interfaces for specifying complex 3D modeling operations remains a major research challenge. Our work focuses on bending and twisting 3D surfaces which have inspired related prior work in both 2D and 3D interfaces [1, 4].

4.1 Bend Gesture

The first interaction we explore deforms a mesh or surface by bending it (see Figure 1 center). The gesture is performed by anchoring the thumb and index finger of one hand to the surface (Figure 2 left hand). These two multi-touch anchor points define the center-line axis of the bend, shown as a solid red line. A finger on the other hand (Figure 2 right hand) is then used as a clutch to lock the bend axis in place. By pivoting the anchored hand around the anchoring fingers, the user bends the mesh either up or down. As the gesture is performed, the user may slide the clutching finger towards or away from the bend axis to vary the curvature of the bend.

The direction and amount of bend to apply to the 3D model is calculated from tracking the movement of the hand above the surface, in this case, using the multi-camera optical tracking system, and interpreting the 3D motion relative to the multi-touch contacts on the surface. On the touch surface, the two points of contact on the anchored hand (thumb and index finger) indicate the axis about which the bend will be applied. By transforming these 2D inputs to the same 3D coordinate system used by the 3D tracking device, we can define a 3D vector that specifies the direction from the thumb touch point to the index finger touch point. We will call this vector $\hat{b} = (b_x, b_y, b_z)$. After the clutching finger from the second hand is engaged, the 3D model will be bent around this axis in proportion to the amount of hand movement about the axis. As the hand is pivoted, the change in orientation relative to its initial pose is recorded and the raw 3×3 rotation matrix is decomposed into an axis of rotation, $\hat{a} = (a_x, a_y, a_z)$, and angle of rotation about that axis, ϕ. The bend angle, θ is calculated as the rotation of the tracker, ϕ, scaled by the projection of the tracker's rotation axis onto the bend axis.

$$\theta = (\hat{a} \cdot \hat{b})\phi \tag{1}$$

The pivoting action of the fingers maps naturally to bending, much the same way a user might bend a piece of paper, while using the display surface as an anchoring surface provides the stability necessary to specify precise changes in the bend angle. The anchored technique integrates seamlessly with more traditional multi-touch inputs, for example, the touch by the clutching hand. This touch point can be moved on the surface during the bending operation to specify curvature, using a style of bi-manual input that is characteristic of many successful multi-touch interfaces.

4.2 Twist Gesture

The second 3D modeling interaction we explore deforms a mesh or surface by twisting it (see Figure 1 right). The gesture is performed using two hands. First one hand is placed (anchored) onto the table on its side (with little finger down and thumb up in the air). Then, the other hand is place on top of the first, again in a sideways orientation. To twist the surface, the hands are spun around the natural axis that forms between them when held in this position.

In this interface, the above-the-surface gestures are captured using the depth camera and algorithm described in Section 3, which outputs the angle of rotation around the up axis for both hands. The motion of the bottom hand specifies the angle of twist for the bottom of the 3D mesh, and the top hand specifies the angle of twist for the top of the 3D mesh. These angles are updated each frame of the interaction based on the relative motion of each hand in the time since the previous frame. The raw rotation data for each frame is converted to an axis of rotation and an angle, which is transformed, as in Equation 1, to calculate the angle of rotation around the z-axis (pointing out of the screen) only. This rotation is then applied to each vertex. For each vertex, the angle to rotate around the z-axis is specified via a smooth linear interpolation between the twist angles specified for the top and bottom of the mesh.

5 Application 2: Anchored 3D Navigation

The second application of anchored multi-touch that we explore is navigation within 3D scenes viewed on stereoscopic multi-touch tables. To date, little work has been done in the area of coupling multi-touch input with stereoscopic displays (notable exceptions are [11, 14]); designing improved interfaces for navigating through 3D datasets remains an important problem in this area of research.

In the following sections, we describe a complete interface for 3D navigation using anchored multi-touch. Traditional 2D gestures (pan, rotate, scale) are integrated into the interface. The anchored gestures address the specific problem of rotating and translating a 3D scene in and out of the table surface. Note that these interactions tend to be performed via indirect gestures, such as mapping an up and down motion on the surface to a translation in and out of the screen [15], when only 2D multi-touch input is available.

5.1 2D Navigation Gestures

Several 2D multi-touch gestures are now commonly used to position, scale, and rotate objects in the plane of a touch surface. We

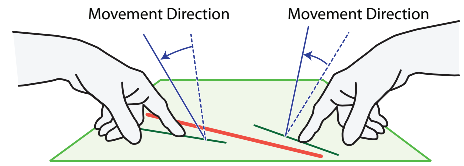

Figure 3: In anchored pivot rotation, the visualization space is rotated by pivoting the hands in a similar direction. The axis of rotation (in red) is derived from the axes of rotation for each hand (in green).

take these as a starting point for the navigation interface described next. Movement of a single touch point translates objects in the xy-plane of the table. Two touch points are used to indicate a rotation about a vertical axis out of the table and scale. Using the familiar metaphor, the points directly under the user's fingers appear to stick to the objects being manipulated during the interaction.

Unlike two-dimensional displays where a scale looks the same as a z-translation from the camera's viewpoint, head-tracked stereoscopic environments are different. A scale action increases the size of the rendered graphics, whereas a z-translation makes them appear as if they are rising up out of the touch surface. For this reason, below, we define a separate anchored gesture for z-translation.

5.2 Extensions for Anchored Pivot Rotation

To specify rotation around the x and y-axes on the table surface, we present a bi-manual interaction. The user touches the surface with a finger tip from each hand. See Figure 3. By keeping the touch point still on the surface (anchored) and pivoting his hands together he can specify a rotation in any direction. The fingers can be thought of as mini joysticks.

To allow direct control, the rotation is limited to have the center of rotation lie on the touch surface, halfway between the fingers, and the rotation axis is constrained to lie within the plane of the table. Allowing explicit control of the center of rotation is important when viewing datasets that do not have an inherent center.

Control over the rotation is also achieved via a filtering mechanism. This gesture uses the motion of both hands, with each hand pivoting in the same direction and by similar amounts. The two motions are averaged to determine the final rotation. For each rendering frame, and for each hand, a vector, $\hat{v}_t$ is created from the touch position on the surface to the tracker position. $\hat{v}_t$ is compared to the previous frame's vector, $\hat{v}_{t-1}$ to find the rotation axis, $\hat{a} = (a_x, a_y, a_z)$ and angle, ϕ.

$$\hat{a} = \hat{v}_t \times \hat{v}_{t-1}, \text{ then projected onto the x-y plane} \tag{2}$$

$$\phi = \arccos(\|\hat{v}_t\| \cdot \|\hat{v}_{t-1}\|) \tag{3}$$

For applications that require extremely precise rotations, we explored the possibility of using a single hand to specify rotations. In this case, the gesture is anchored with the thumb and index finger of one hand, in a similar posture to the bend gesture described previously. The line on the surface connecting the two touch points is used as the rotation axis. This enables the user to precisely specify the axis of rotation and provides additional anchoring support to control rotations. The disadvantage relative to the bi-manual interaction is that the bi-manual gesture can be more ergonomic for rotations toward or away from the body. Also, the hands can be placed arbitrarily far apart in the bi-manual case, with is useful for limiting occlusions when stereoscopic displays are used.

One challenge in designing above-the-surface interfaces is distinguishing 2D from 3D gestures while still maintaining modeless operation. Our solution uses a series of checks to determine whether the hands are pivoting around their anchors and whether the motions of the two hands are moving in the same direction

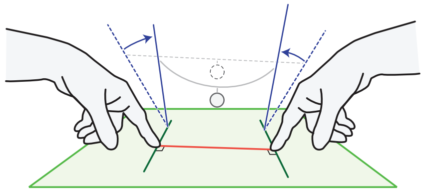

Figure 4: Illustration of the weighted string metaphor for z-translation. Pivoting the hands towards each other lowers the weight

by roughly the same amount (correlated) or moving in opposite directions (anti-correlated). To identify pivoting motion, the total amount of movement over a sliding time window is calculated. In practice, we found that touches moving less than 0.06 in within a window size of 0.06 sec worked well for distinguishing the traditional 2D multi-touch inputs describe above from the anchored gestures described in this section. For a motion to be correlated, the hands must be moving in the same direction and by the same amount, causing the rotation axes and angles from each hand to be similar. A simple threshold is sufficient to test this. We found that 70 degrees for rotation axes and 20 degrees for rotation angles worked well for a majority of our users. Although users quickly get better at moving their hands in the same direction with some practice, at first they have a tendency to pivot their hands slightly towards each other (the reason for the higher rotation axis threshold), particularly when rotating away from their body. We also found it useful to limit the rotation axis from changing during an anchored rotation operation unless the difference between the new axis and the current one is more than 20 degrees.

5.3 Extensions for Anchored Z-Translation

Translating objects into and out of the plane of the touch table is a difficult problem for multi-touch. Since this interaction occurs in a 3D stereoscopic environment, z-translation is distinct from simple zooming. We introduce an anchored gesture metaphor for z-translation that is inspired by the successful balloon selection technique [2], in which touches are imagined to control a string on a virtual balloon. In our case, with one finger from each hand touching the surface, picture a string running from one hand to the other (Figure 4.) A metaphorical weight in the center of the string is attached to the visualization space. Pivoting the hands towards each other creates slack in the string, causing the weight and the visualization space to drop lower (translating in the negative z direction). Pivoting the hands away from each other tightens the string bringing the weight and visualization space back up.

Although the motion in both this gesture and the rotation gesture described previously is to pivot the hands around a single touch point, they are distinguished by the correlation or anti-correlation of the direction of movement. For z-translation, we check that the hands are rotating in opposite directions by making sure that the rotation axes for each hand are pointing in opposite directions. In practice, we require that the difference between rotation axes for each hand be greater than 120 degrees. We also require that the direction of hand motion be along the line from one touch point to the other. When this is the case, the rotation axes for each hand will be close to perpendicular with this line. This is tested by requiring the angle difference between rotation axes for each hand and the line between touch points be within 70 and 120 degrees.

6 Conclusion and Future Work

We have presented two applications exploring possible uses of these types of interaction. During informal use, anchored gestures were greeted with enthusiasm by users, who thought they have the potential to make multi-touch interaction even more intuitive. However, to be successful, designers must be cautious to make sure the gestures are ergonomic and self-revealing. We believe that, as multi-touch technology becomes more ubiquitous, the advantages of the passive haptic feedback and stability that anchored gestures provide will become even more useful, and we are excited by the potential to expand the impact of multi-touch, especially for 3D applications.

Acknowledgements

This work was supported in part by the University of Minnesota Office of the Vice President for Research, Grant-in-Aid of Research, Artistry, and Scholarship Program and National Academies Keck Futures Initiative.

References

[1] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and K. Singh. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 111–118, 1999.

[2] H. Benko and S. Feiner. Balloon selection: A multi-finger technique for accurate low-fatigue 3D selection. In *IEEE Symposium on 3D User Interfaces (3DUI) 2007*, 2007.

[3] H. Benko, E. Ishak, and S. Feiner. Cross-dimensional gestural interaction techniques for hybrid immersive environments. In *Proceedings of IEEE Virtual Reality, 2005*, pages 209 –216, 2005.

[4] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 183–188, 1992.

[5] T. Grossman, D. Wigdor, and R. Balakrishnan. Multi-finger gestural interaction with 3D volumetric displays. *ACM Transactions on Graphics*, 24:931–931, 2005.

[6] C. Harrison, H. Benko, and A. D. Wilson. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 441–450, 2011.

[7] O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz. Interactions in the air: Adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM Symposium on User Interface Software and Technology*, pages 139–148, 2009.

[8] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24:1134–1141, 2005.

[9] R. S. Kattinakere, T. Grossman, and S. Subramanian. Modeling steering within above-the-surface interaction layers. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 317–326, 2007.

[10] N. Marquardt, R. Jota, S. Greenberg, and J. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TCI3 Conference on Human Computer Interaction - INTERACT 2011*, 2011.

[11] F. Steinicke, K. H. Hinrichs, J. Schöning, and A. Krüger. Multi-touching 3D data: Towards direct interaction in stereoscopic display environments coupled with mobile devices. In *Advanced Visual Interfaces (AVI) Workshop on Designing Multi-Touch Interaction Techniques for Coupled Public and Private Displays*, pages 46–49, 2008.

[12] S. Subramanian, D. Aliakseyeu, and A. Lucero. Multi-layer interaction for digital tables. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 269–272, 2006.

[13] Y. Takeoka, T. Miyaki, and J. Rekimoto. Z-touch: an infrastructure for 3D gesture interaction in the proximity of tabletop surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 91–94, 2010.

[14] D. Valkov, F. Steinicke, G. Bruder, and K. H. Hinrichs. 2D touching of 3D stereoscopic objects. In *ACM Proceedings of CHI 2011 Conference on Human Factors in Computing Systems*, 2011.

[15] L. Yu, P. Svetachov, P. Isenberg, M. Everts, and T. Isenberg. FI3D: Direct-touch interaction for the exploration of 3D scientific visualization spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613 –1622, 2010.