Bimanual Marking Menu for Near Surface Interactions

François Guimbretière
Information Science Department
Cornell University
Ithaca NY, 14850
francois@cs.cornell.edu

Chau Nguyen
Computer Science Department
Cornell University
Ithaca NY, 14850
cnguyen@cs.cornell.edu

ABSTRACT

We describe a mouseless, near-surface version of the Bimanual Marking Menu system. To activate the menu system, users create a pinch gesture with either their index or middle finger to initiate a left click or right click. Then they mark in the 3D space near the interactive area. We demonstrate how the system can be implemented using a commodity range camera such as the Microsoft Kinect, and report on several designs of the 3D marking system.

Like the multi-touch marking menu, our system offers a large number of accessible commands. Since it does not rely on contact points to operate, our system leaves the non-dominant hand available for other multi-touch interactions.

Author Keywords

Near surface interactions; Range camera; Bimanual marking menu.

ACM Classification Keywords

H.5.2 [Information Interfaces And Presentation]: User Interfaces - Interaction styles;

INTRODUCTION

With the wide availability of multi-touch surfaces, we have seen a revival of gesture-based interfaces based on this technology [4, 20, 21, 23]. The two prevalent tracking technologies (capacitive and optical tracking) are focusing on contact interactions. Although optical tracking technology can be modified to allow for interactions above the surface [2], interactions in free space may cause fatigue during long periods of use. Following the example of hover interfaces in pen-computing [7], we are exploring near-surface interactions, which take place a couple of inches above the surface. Because these interactions can be performed with the users' arms resting on the surface, they are better adapted to long periods of use. To explore this design space, we built an indirect tracking pad using a Microsoft Kinect, a commodity depth of field camera scanning a hand through a piece of acrylic (Figure 1). Using this system, we implemented a mouseless, near surface version of the Bimanual Marking Menu technique [16].

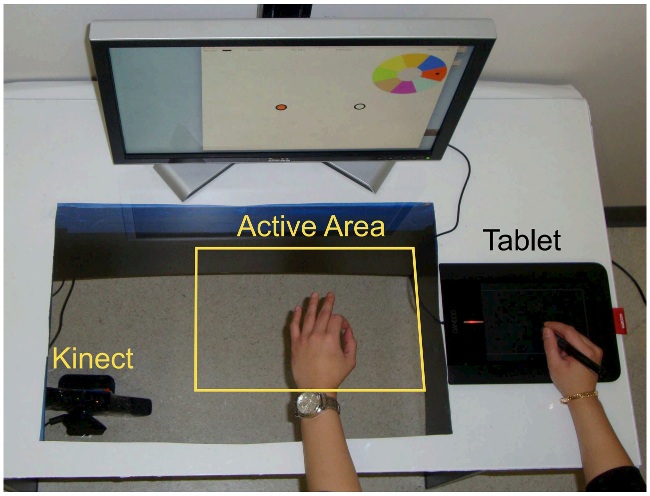

Figure 1: The setting of our interactive pad. The Kinect scans the hand through a hole covered with a piece of acrylic

As an invocation mechanism, we extended Wilson's pinching interaction [22] to take into account depth data and detect which finger is creating the pinch. This lets us simulate multi-button mouse interactions. We then explored the possibility of significantly increasing the number of markings by allowing users to mark in the 3D space just above the interaction surface. In the *layered* setting, users are marking on three different levels that are each parallel to the resting surface. In the *spherical* setting, an approach similar to C^3 [6] and Grossman et al.'s Volumetric display system [8], users are marking directly in 3D, both upward and downward. In the *hemispherical setting*, users are also marking in 3D but markings are flat or directed upward. The 3 designs have a similar number of available markings (24, 26 and 24 respectively) but different pros and cons we report on. Together, these extensions offer access to a very large number commands without introducing any devices or contact points for the non-dominant hand. This leaves this hand free to participate in other gesture-based activities.

PREVIOUS WORKS

Our work draws on prior extensions of the original Marking Menu system [10]. Several systems have demonstrated the use of marking menu in 3D to interact in virtual environments [6], to control devices from a distance [1, 11, 15], and interact with Volumetric displays [8]. While these approaches are considering free space interaction, we are studying near surface interaction for which both arms are resting on the interactive surface. Further with the exception of Lenman et al. [11] and Bally et al. [1] these

CHI 2012, May 5-10, 2012, Austin, TX, USA.

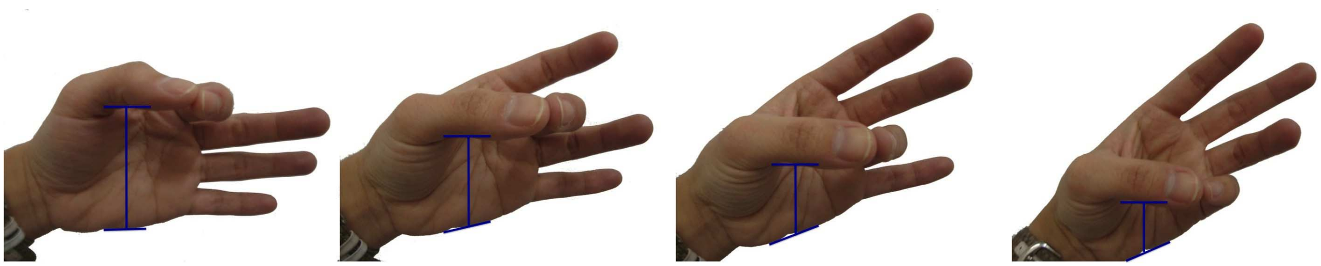

Figure 2: Multi-finger pinching showing the different poses of the hand depending on which finger is pinched

systems require users to wear a hand tracking system while we are focusing on bare-hand interactions using a Kinect.

Odel's bimanual marking menu (BMM) [16] is an extension of the marking menu [10] in which the marking is performed by the non-dominant hand. This setting makes it a very efficient bimanual technique as shown by Chen et al. [3]. The original implementation relies on a simple mouse for marking. This makes BMM somewhat unpractical for table-top surfaces because it forces users to switch between the mouse and bare handed interactions. One solution might be to use the SDMouse proposed by Matejka et al. [14], but our approach offers a larger number of directly accessible marks. Another alternative might be to use Lepinski et al.'s multi-touch marking menu system [12]. Our approach is an alternative solution to provide a large number of marks. It does not conflict with other multi-touch gestures, can be extended to parameter entry in 3D space and relies on a more relaxed posture for the marking hand.

NEAR SURFACE BMM

Our setting, shown in Figure 1, is based on a Kinect placed under a table that tracks the non-dominant hand above the table through a transparent window. To the right of the window is a Wacom tablet used to track pen interactions performed by the dominant hand. While the system described here could be used to mark with the dominant hand, BMM is very well adapted to the capabilities of our Kinect-based prototype. The Kinect is a somewhat low-resolution sensor (about 18dpi on the surface of the table). As such, it seems best to focus on marking interactions performed by the non-dominant hand, while leaving touch or pen interactions for the dominant hand.

We replace the multi-button mouse by combining Wilson's pinching system [22] with depth data provided by the Kinect. Depth data provide two important pieces of information: 1) they let us detect which finger is forming a pinch with the thumb; 2) they let us track the hand in 3D space allowing for marks in 3 dimensions.

A multi-button pinch using the Kinect

To simulate a multi-button mouse, we use the 3D information provided by the Kinect to detect which finger is forming a pinch with the thumb (similar to the PinchWatch [13] design). This is possible because, while the thumb is able to move in front of each finger, each finger has relatively little leeway to move up or down. As a result, the plane defined by each finger pinch is very distinctive (Figure 2).

To activate the menu system, the user performs a pinch gesture with his or her non-dominant hand. The user then moves the hand towards the direction of the desired menu command, and finally releases the pinch to finish marking. Our system includes a dead-zone of about 30mm, and feedback is provided after a 500ms timeout. A typical gesture is 45 mm long.

Implementation

We initially expected that implementing the pinching interaction with the Kinect would be fairly simple since the hand would appear much closer than any other object visible through the window. Unfortunately, the complex shape of a pinching hand creates a set of shadows in the tracking pattern (Figure 3). This in turn creates "holes" in the tracking data returned by the camera which makes simple 3D thresholding unreliable.

Upon receiving the depth data from the Kinect, our system uses PrimeSense NITE library to detect a hand entering the frame. It then limits further processing to a 200 pixel wide bounding box around the hand. Next, we apply Felzenszwalb et al.'s [5] segmentation algorithm on the depth map to identify possible holes and reject holes with skin color [19]. If a hole is detected, we proceed to compute the hole's plane by using a least square plane-fitting on the points at the border of the hole. To normalize the data, we then re-project the point cloud so that the plane's normal becomes the vertical axis. In this coordinate system, we

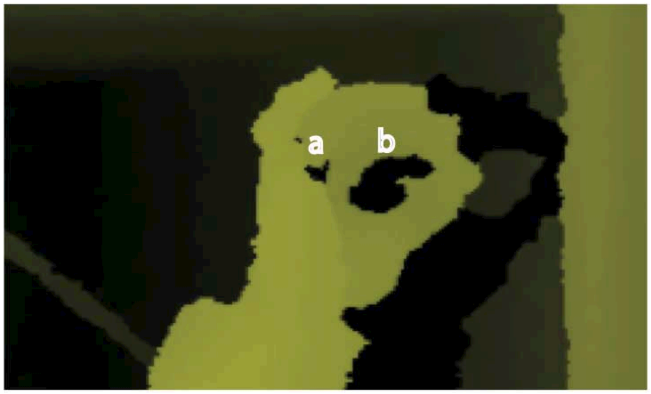

Figure 3: Multiple shadows created by the hand during pinching. a) self shadow within the hand; b) shadow of the hand seen through the pinch

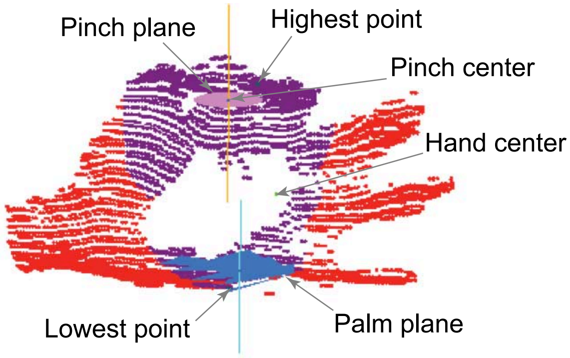

Figure 4: Raw parameters to be used by the Support Vector Machine for an index pinch. Only purple points are considered for computing the highest and lowest point.

compute the plane defined by the bottom of the palm, the points with maximum and minimum height inside a cylindrical bounding box centered on the hole, the middle of the hand, and the middle of the hole (Figure 4). These parameters are used as features to infer which finger is forming the pinch using a Support Vector Machine multiclass classifier [18]. We observed 93% recognition rate for the index pinch and 89% for the middle finger. We also considered pinching with the ring and pinky, but the recognition rate was too low and pinching with the pinky proved uncomfortable.

Near surface 3D marking

Since we receive 3D information from the camera, we can track 3D gestures. We focused on near surface interactions for which the forearms can comfortably rest on the table during marking. Importantly, the hand can still perform small vertical movements in this context. We considered several different approaches for marking in near space. The *layered* design was inspired by hover interactions [7]. In that case, a different set of menus is called depending on the height at which the marking is performed (Figure 5 left column). Three levels (each about one inch apart) offered a good compromise between the size of hand movements and robust detection. Assuming a detection system differentiating between two pinch types (index and middle finger), this allowed us to access 48 commands with a single mark (3 layers * 8 directions * 2 pinch types).

Another approach is to consider marking in 3D instead of 2D [6, 8]. In this *spherical* approach, one can mark in one of 26 directions accessible by combining compass directions with up and down gestures (Figure 5, middle). When differentiating between two pinch types, users can access up to 52 commands in a single marking. Noting that marking downward might be awkward, we considered a *hemi-spherical* scheme, in which marking is performed either flat, at a shallow upward angle (~30°), or at a steeper upward angle (>45°, Figure 5 right column). Assuming two pinch types, this yields 48 commands in a single mark. For all layouts, the number of accessible commands can be significantly increased by considering multi-marks hierarchical menus [24].

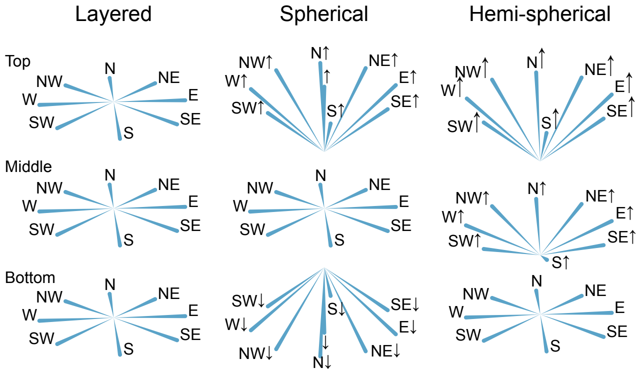

Figure 5: Three different settings for 3D marking in near space. *Left*: layered approach; *Middle*: spherical approach similar to C^3 [6] and Grossman et al. system [8]; *Right*: 3-level hemispherical approach.

DISCUSSION

We now report specific observations made in fine-tuning our marking system.

Sensor limitations

While the Kinect served well as an inexpensive prototyping tool, its main limitation is its resolution. This curtails reliability in detecting the pinch gesture. Based on an analysis of video footage, we discovered that the system detects a pinch earlier than perceived by users. This leads to longer stroke marking time, but may not influence total performance time since the extra marking time occurs during the preparation phase.

Another important problem is the default latency introduced by the processing time of the Kinect, which we estimate at about 150ms. At present, this limits the overall response time of our interface, although a more powerful image processing system would address this concern.

The placement of the camera is very important. In our implementation we had to place the Kinect below and to the left of the tracking area to allow for a clear view of the pinch gesture when the hand was in a natural resting position (see Figure 1). This setting also limited glare problems observed when the Kinect was placed below the mid-point of the window in the table. We believe that a custom designed optical system would allow us to address most of the occlusions problem we observed.

3D marking

Among the 3 marking solutions described above, the *layered* approach (Figure 5, left column) was found to be the most reliable in informal testing. As expected, the *spherical* approach was awkward with respect to downward markings, since one has to remember to raise the hand above the surface before marking down. The *hemi-spherical* design was unreliable since it is difficult to gauge the proper angle of a mark. Instead one relies on the height of the hand at the end of the mark. This approach negates the scale invariance which is key to Marking Menu [10] performance. Empirical evaluation will be needed to characterize these problems.

FUTURE WORK AND CONCLUSION

As a next step, we are planning to conduct an empirical evaluation of our system to compare its overall performance with systems such as the Multi-Touch Marking Menu [12]. We are also planning to explore how it can be extended to implement versions of FlowMenu [9] and control menu [17] allowing the non-dominant hand to control one parameter with 6 degrees of freedom after completing command selection.

We presented an extension of the Bimanual Marking Menu in which users mark in 3D using a pinch gesture of their non-dominant hand. We implemented our system using an off-the-shelf Kinect camera and demonstrated that we can reliably detect two different pinch types and up to 52 different markings. Our preliminary findings indicate that assuming a faster camera, the system has the potential to be a viable option for selecting a large number of commands on a multi-touch surface. Given these characteristics, our approach can aid in the implementation of complex applications on multi-touch surfaces.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant IIS-0936105. Shenwei Liu provided the template for the drawing application we used for our prototype. We thank Corinna Löckenhoff for comments on an earlier draft of this paper.

REFERENCES

1. Bailly, G., R. Walter, r. M, ller, T. Ning, and E. Lecolinet. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. *Proceedings of INTERACT'11*, pp. 248 - 262.
2. Benko, H. and A.D. Wilson, DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface, Microsoft Research, MSR-TR-2009-23, 2009.
3. Chen, N., F. Guimbretière, and C. Loeckenhoff, Relative Role of Merging and Two Handed Operation on Command Selection Speed. *International Journal of Human-Computer Studies*, 2008. **66**(10): p. 729 - 740.
4. Dang, C.T., M. Straub, and E. André. Hand distinction for multi-touch tabletop interaction. *Proceedings of ITS'09*, pp. 101 - 108.
5. Felzenszwalb, P.F. and D.P. Huttenlocher, Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 2004. **59**(2): p. 167 - 181.
6. Grosjean, J., J.-M. Burkhardt, S. Coquillart, and P. Richard. Evaluation of the Command and Control Cube. *Proceedings of ICMI'02*, pp. 473 - 479.
7. Grossman, T., K. Hinckley, P. Baudisch, M. Agrawala, and R. Balakrishnan. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. *Proceedings of CHI'06*, pp. 861 - 870.
8. Grossman, T. and R. Balakrishnan. Collaborative interaction with volumetric displays. *Proceedings of CHI'08*, pp. 383 - 392.
9. Guimbretière, F. and T. Winograd. FlowMenu: combining command, text, and data entry. *Proceedings of UIST'00*, pp. 213 - 216.
10. Kurtenbach, G., The Design and Evaluation of Marking Menus. *PhD thesis, University of Toronto*, 1993.
11. Lenman, S., L. Bretzner, and B. Thuresson. Using marking menus to develop command sets for computer vision based hand gesture interfaces. *Proceedings of NordiCHI'02*, pp. 239 - 242.
12. Lepinski, G.J., T. Grossman, and G. Fitzmaurice. The design and evaluation of multitouch marking menus. *Proceedings of CHI'10*, pp. 2233 - 2242.
13. Loclair, C., S. Gustafson, and P. Baudisch. PinchWatch: A Wearable Device for One-Handed Microinteractions. *Proceedings of MobileHCI'10, Workshop on Ensembles of On-Body Devices*, pp. 1 - 4.
14. Matejka, J., T. Grossman, J. Lo, and G. Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. *Proceedings of CHI'09*, pp. 1073 - 1082.
15. Ni, T., R.P. McMahan, and D.A. Bowman. rapMenu: Remote Menu Selection Using Freehand Gestural Input. *Proceedings of 3DUI'08*, pp. 55 - 58.
16. Odell, D., R. Davis, A. Smith, and P.K. Wright. Toolglasses, Marking Menus, and Hotkeys: A Comparison of One and Two-Handed Command Selection Techniques. *Proceedings of GI'04*, pp. 17 - 24.
17. Pook, S., E. Lecolinet, G. Vaysseix, and E. Barillot. Control menus: excecution and control in a single interactor. *Proceedings of CHI'00 Extended Abstracts*, pp. 263 - 264.
18. Tsochantaridis, I., T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. *Proceedings of ICML'04*, pp. 104 - 111.
19. Vijayanandh, R. and G. Balakrishnan. Human face detection using color spaces and region property measures. *Proceedings of ICARCV'10*, pp. 1605 - 1610.
20. Wang, F., X. Cao, X. Ren, and P. Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. *Proceedings of UIST'09*, pp. 23 - 32.
21. Wang, F. and X. Ren. Empirical evaluation for finger input properties in multi-touch interaction. *Proceedings of CHI'09*, pp. 1063 - 1072.
22. Wilson, A.D. Robust computer vision-based detection of pinching for one and two-handed gesture input. *Proceedings of UIST'06*, pp. 255 - 258.
23. Wobbrock, J.O., M.R. Morris, and A.D. Wilson. User-defined gestures for surface computing. *Proceedings of CHI'09*, pp. 1083 - 1092.
24. Zhao, S. and R. Balakrishnan. Simple vs. compound mark hierarchical marking menus. *Proceedings of UIST'04*, pp. 33 - 42.