Ray-Casting Based Interaction Using an Extended Pull-Out Gesture for Interactive Tabletops

Shun Fujita
fujita@iplab.cs.tsukuba.ac.jp
University of Tsukuba, Japan

Naoki Yanagihara
yanagihara@iplab.cs.tsukuba.ac.jp
University of Tsukuba, Japan

Buntarou Shizuki
shizuki@cs.tsukuba.ac.jp
University of Tsukuba, Japan

Shin Takahashi
shin@cs.tsukuba.ac.jp
University of Tsukuba, Japan

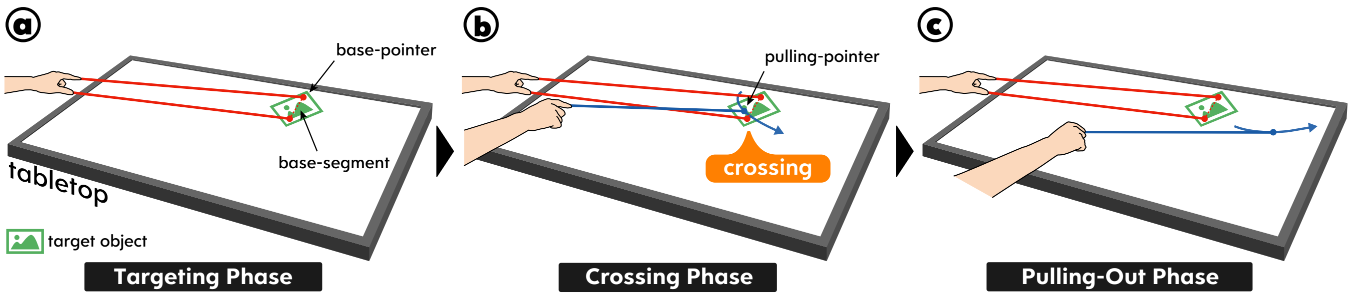

Figure 1: Procedure of an *extended pull-out gesture.* A user performs this gesture in three phases. During the targeting phase, the user moves two ray-casting points projected from two fingers of the non-dominant hand (*base-pointers*) in such a manner that the midpoint of the segment joining the two base-pointers (*base-segment*) lies on a target object (a). During the crossing phase, the user crosses the base-segment (b) using the ray-cast point projected from a finger of the dominant hand (*pulling-pointer*). During the pulling-out phase, the user manipulates the object by moving the base-pointers and the pulling-pointer (c).

ABSTRACT

In this paper, we present a method that allows the users to manipulate a remote object on tabletops using an *extended pull-out gesture.* The user performs the gesture by moving a ray-cast point projected from the dominant hand between two ray-cast points projected from the non-dominant hand. This allows the user to manipulate a remote object on a tabletop in a similar manner to manipulating a close object. We implemented a test system to the feasibility of our method and we showed applications to manipulate a remote object.

CCS CONCEPTS

• **Human-centered computing → User interface design**; *Graphical user interfaces*; Pointing; Gestural input.

KEYWORDS

crossing, mid-air interaction, remote interaction, bimanual

OZCHI'19, December 2–5, 2019, Fremantle, WA, Australia

ACM ISBN 978-1-4503-7696-9/19/12.
https://doi.org/10.1145/3369457.3369528

ACM Reference Format:
Shun Fujita, Naoki Yanagihara, Buntarou Shizuki, and Shin Takahashi. 2019. Ray-Casting Based Interaction Using an Extended Pull-Out Gesture for Interactive Tabletops. In *31ST AUSTRALIAN CONFERENCE ON HUMAN-COMPUTER-INTERACTION (OZCHI'19), December 2–5, 2019, Fremantle, WA, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3369457.3369528

1 INTRODUCTION

A direct touch on a tabletop has a problem that a user cannot manipulate an object located outside a user's reach (hereafter referred to as a *remote object*) without moving closer to it. More specifically, Toney and Thomas [15] reported that over 90% of direct touches are constrained to a region extending up to 34 cm deep in front of the user.

For this reason, several methods have been proposed for manipulation of remote objects [1, 3, 11, 12, 14]. These methods expand the user's reach virtually, enabling the user to manipulate a remote object (e.g., select, translate, rotate, scale, or all of them) without moving any closer to it. However, they are not designed to provide the user with command execution on a remote object, such as menu selection or text entry on a remote text box. In contrast, systems developed in other papers [5, 8] allow the user to execute such commands on a remote object, although they require finger movements that are different from the ones used to manipulate an object within the user's reach (*close object*) via direct touch, requiring the user to learn an additional method to use the system.

In this paper, we present a method that allows the user to manipulate a remote object on a tabletop (Figure 1). It adopts a ray-casting based interaction employing an *extended pull-out gesture*. The original pull-out gesture [17] is a bimanual multi-touch gesture utilizing crossing [2] — the user performs it by moving a touch point corresponding to the dominant hand between two touch points corresponding to the non-dominant hand, which invokes a widget containing commands on a tabletop. In contrast, the user performs the *extended pull-out gesture* by moving a ray-cast point (i.e., the intersection of a ray cast from a finger with the surface of the tabletop) projected from the dominant hand between two ray-cast points projected from the non-dominant hand. The user can perform these gestures with pointer movements that are similar to the usual finger movements required to manipulate a nearby area via direct touch.

2 RELATED WORK

Several methods to manipulate a remote object on a tabletop have already been proposed. Parker et al. [12] presented a ray-casting based method for selecting a remote object on a tabletop using a stylus. Methods using multi-touch interaction have explored various metaphors including a virtual mouse [5], a grabbing metaphor [1], a portal metaphor [16], and a telescope of a remote area [8] to manipulating a remote object or transferring a remote object to a nearby area. Newn et al. [11] proposed a method of manipulating a remote object by combining direct touch and gaze operations that considered the characteristics of a user's gaze onto a tabletop. Among the methods using mid-air interaction, Pointable proposed by Banerjee et al. [3], combines perspective-based pointing using the index finger of the dominant hand with rotation and scaling operations via movement of the non-dominant hand. Moreover, Tochihara et al. [14] proposed a method utilizing multiple pointers that can adjust a control-display (C-D) ratio with the heights of the user's hands from a tabletop.

Multi-pointer techniques using mid-air hand gestures also have also been explored to manipulate large vertical displays [4, 10]. Banerjee et al. [4] compared some remote pointing techniques performed using laser pointers and user's fingers, in single and multi-finger pointing interactions. Matulic et al. [10] evaluated the performance of a multi-finger ray-casting design space, in which each finger projects a ray onto a large screen, which forms pointers on the screen. The user can use these pointers to manipulate the display from a distance. The authors also proposed the use of positional relations for the pointers, which form 2 dimentional geometric shapes (e.g., circle and rectangle), to execute various commands.

Bezerianos et al. [6] proposed the vacuum, which employs a circular widget that moves remote objects closer to itself. Further, Stellmach et al. [13] proposed a method that manipulates an object on a distant display by combining the user's gaze and head movements with direct touches on smartphones.

The *extended pull-out gesture* presented in this paper is a mid-air gesture, which uses rays emitted from the user's fingers similar to [10]. However, in contrast to the aforementioned methods, our method focuses on allowing the user to manipulate a remote object on a tabletop by using finger movements that are similar to those used to manipulate a nearby object.

3 RAY-CASTING BASED INTERACTION USING AN EXTENDED PULL-OUT GESTURE

The titular method of this paper allows the user to manipulate a remote object on a tabletop with the help of a ray-casting based interaction using an *extended pull-out gesture*.

3.1 Extended Pull-Out Gesture

The user performs the *extended pull-out gesture* by moving ray-cast points (i.e., the intersection of a ray cast from a finger with the surface of the tabletop) according to the following procedure (Figure 1):

a) During the targeting phase, the user moves two ray-casting points projected from two fingers of the non-dominant hand (*base-pointers*) in such a manner that the midpoint of the segment joining the two base-pointers (*base-segment*) lies on the target object.
b) During the crossing phase, the user crosses the base-segment using a ray-cast point projected from a finger of the dominant hand (*pulling-pointer*).
c) During the pulling-out phase, the user manipulates the object by moving the base-pointers and the pulling-pointer.

In the original pull-out gesture [17], which was designed for touch-based interaction, the user could cancel a gesture by lifting fingers of the non-dominant hand from the tabletop. However, the user cannot cancel the *extended pull-out gesture* by same manner since rays are permanently emitted from fingers. Hence, the user can cancel a gesture by moving the base-pointers closer with each other than a threshold.

3.2 Merits

This design allows the user to manipulate a remote object by using an *extended pull-out gesture* in a manner that is similar to the manipulation of a close object using the original pull-out gesture, as the pointer movements in both cases are identical.

The *extended pull-out gesture* has variants similar to those of the original one. For example, the number of crossings, crossings with multiple pulling-pointers or expansion of the base-segment to other geometrical shapes can be considered [9, 10, 17]. Such variants allow the assignment of various functions to the pull-out gestures.

Moreover, both the extended and the original pull-out gestures use distinctive finger movements. Thus, those gestures are unlikely to occur unintentionally or conflict with other touch and mid-air gestures. Consequently, the *extended pull-out gesture* has the potential to coexist with other systems of gestures.

4 APPLICATIONS

In this section, we present applications of the *extended pull-out gesture*.

4.1 Pull-Out Menu

A pull-out menu is a menu whose size and orientation are adjustable that can be invoked by an *extended pull-out gesture*. While pulling out the menu, the users can adjust the menu size by varying the length of the base-segment and orientation by varying the direction of the pulling pointer with respect to the midpoint of the base-segment (Figure 2). When the menu is fully pulled-out, its size becomes fixed, and its position follows the position of the base-segment. An item in the pulled-out menu can be selected by the crossing the pulling-pointer.

This design enables each user surrounding a tabletop to adjust the menu orientation and size for ease of manipulation. Moreover, in situations in which the user accesses a menu for a remote object, the user can move the menu closer by moving the base-pointers.

4.2 Object Translation, Rotation, and Scaling

An *extended pull-out gesture* can be used for translation, rotation, and scaling operations. When a user performs an *extended pull-out gesture* on an object, it follows the translation, rotation, and scaling of the base-segment. The user can terminate this operation by recrossing the base-segment with the pulling-pointer.

4.3 Pull-Out Keyboard

A user can invoke a pull-out keyboard by performing an *extended pull-out gesture* on a text box (Figure 3a) to enter a text in the text box. The user can move the keyboard simply by moving the pointers. The keyboard can be pinned by touching and releasing the fingers on a tabletop (Figure 3b). The pinned keyboard functions as a software keyboard that can be typed on using touch, and is closed by pressing the close button.

5 TEST SYSTEM

To demonstrate the viability of our method, we implemented a test system consisting of a motion capture system (Optitrack Flex 3; 6 cameras) to track the user's fingers and a tabletop display (Panasonic TH-50PH12). Retroreflective markers were attached to the user's wrists and fingers, as shown in Figure 4 to carry out the demonstration For the test implementation, the number of rays was restricted to three — one ray cast from the dominant hand and two rays cast from the non-dominant hand to stabilize the tracking as the number of available cameras was limited. The ray cast from each finger was engineered to be emitted from the metacarpophalangeal joint of the finger in the direction of its fingertip, identical to the set-up developed in [10].

The system applied 1 € filters [7] to reduce the jitter caused by trembling of hands and fingers. Specifically, the system dynamically increased the coefficients of the filters when the length of the ray increased or when the angle between the ray and the display decreased because the jitter of a pointer tends to become larger. This also contributes to decreased delays in pointer movement when the user manipulates a nearby area.

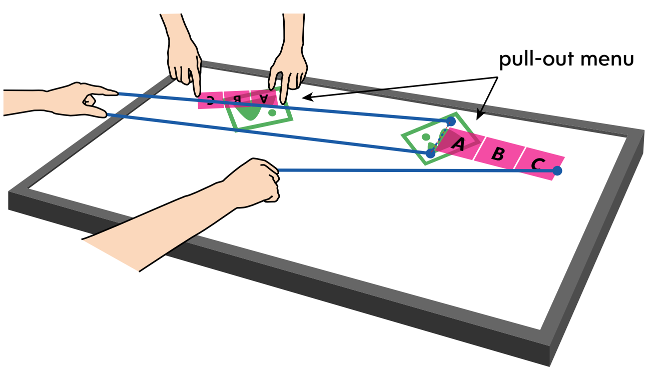

Figure 2: Pull-out menus. Users surrounding a tabletop can pull out menus simultaneously. Each user can adjust the size and orientation of the menus to enable comfortable use from the user's position.

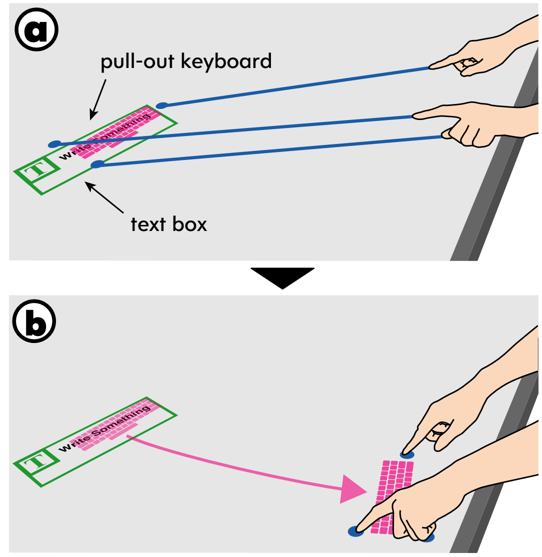

Figure 3: A pull-out keyboard. a) A user invokes a keyboard on a text box. b) The user pulls in a keyboard and pins it.

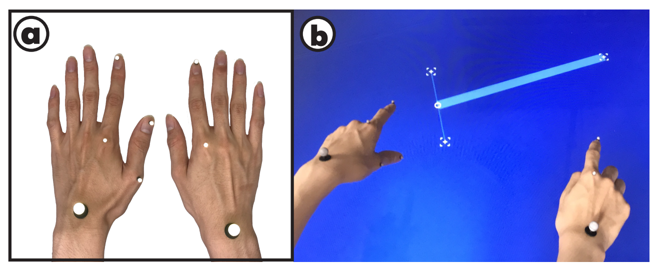

Figure 4: Test system. a) Marker arrangements on the hands and fingers. b) A user performs a pull-out gesture using the test system.

6 CONCLUSION AND FUTURE WORK

We presented a method that allows the user to manipulate a remote object on a tabletop using an *extended pull-out gesture*. The user performs the gesture by moving a ray-cast point projected from the dominant hand between two ray-cast points projected from the non-dominant hand. Our method provides the user to manipulate a remote object and close object in a similar manner. We implemented a test system to confirm the feasibility of our method and exhibited possible applications using the extended pull-out gesture, e.g. the pull-out menu and the pull-out keyboard.

We plan to conduct user studies to evaluate the performance of the *extended pull-out gesture* and its applications in the future. Further, we plan to test the feasibility of our method in different situations and with different situations, such as large vertical displays. Additionally, we plan to develop a version using a markerless motion capture system to make the implementation of our method more practical.

REFERENCES

[1] Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. 2009. I-Grabber: Expanding Physical Reach in a Large-Display Tabletop Environment Through the Use of a Virtual Grabber. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, 61–64. https://doi.org/10.1145/1731903.1731917

[2] Johnny Accot and Shumin Zhai. 2002. More Than Dotting the I's — Foundations for Crossing-based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 73–80. https://doi.org/10.1145/503376.503390

[3] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2011. Pointable: An In-air Pointing Technique to Manipulate Out-of-reach Targets on Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 11–20. https://doi.org/10.1145/2076354.2076357

[4] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2012. MultiPoint: Comparing Laser and Manual Pointing as Remote Input in Large Display Interactions. *International Journal of Human-Computer Studies* 70, 10 (2012), 690–702. https://doi.org/10.1016/j.ijhcs.2012.05.009 Special issue on Developing, Evaluating and Deploying Multi-touch Systems.

[5] Tom Bartindale, Chris Harrison, Patrick Olivier, and Scott E. Hudson. 2011. SurfaceMouse: Supplementing Multi-Touch Interaction with a Virtual Mouse. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '11)*. ACM, New York, NY, USA, 293–296. https://doi.org/10.1145/1935701.1935767

[6] Anastasia Bezerianos and Ravin Balakrishnan. 2005. The Vacuum: Facilitating the Manipulation of Distant Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 361–370. https://doi.org/10.1145/1054972.1055023

[7] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1€ Filter: A Simple Speed-Based Low-pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

[8] Takuro Kuribara, Takuto Yoshikawa, Buntarou Shizuki, and Jiro Tanaka. 2014. Handyscope: A Remote Control Technique Using Pull-Out Gesture. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 175–176. https://doi.org/10.1145/2559206.2579477

[9] Yuexing Luo and Daniel Vogel. 2014. Crossing-Based Selection with Direct Touch Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2627–2636. https://doi.org/10.1145/2556288.2557397

[10] Fabrice Matulic and Daniel Vogel. 2018. Multiray: Multi-Finger Raycasting for Large Displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 245, 13 pages. https://doi.org/10.1145/3173574.3173819

[11] Joshua Newn, Eduardo Velloso, Marcus Carter, and Frank Vetere. 2016. Multimodal Segmentation on a Large Interactive Tabletop: Extending Interaction on Horizontal Surfaces with Gaze. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 251–260. https://doi.org/10.1145/2992154.2992179

[12] J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. 2005. TractorBeam: Seamless Integration of Local and Remote Pointing for Tabletop Displays. In *Proceedings of Graphics Interface 2005 (GI '05)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 33–40. http://dl.acm.org/citation.cfm?id=1089508.1089515

[13] Sophie Stellmach and Raimund Dachselt. 2013. Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 285–294. https://doi.org/10.1145/2470654.2470695

[14] Naoya Tochihara, Toshiki Sato, and Hideki Koike. 2016. A Remote Pointing Method with Dynamic C-D Ratio During a Pinching Gesture for Large Tabletop Systems. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 553–559. https://doi.org/10.1145/2851581.2892579

[15] Aaron Toney and Bruce H. Thomas. 2006. Applying Reach in Direct Manipulation User Interfaces. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI '06)*. ACM, New York, NY, USA, 393–396. https://doi.org/10.1145/1228175.1228255

[16] Simon Voelker, Malte Weiss, Chat Wacharamanotham, and Jan Borchers. 2011. Dynamic Portals: A Lightweight Metaphor for Fast Object Transfer on Interactive Surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 158–161. https://doi.org/10.1145/2076354.2076384

[17] Takuto Yoshikawa, Buntarou Shizuki, and Jiro Tanaka. 2012. HandyWidgets: Local Widgets Pulled-Out from Hands. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*. ACM, New York, NY, USA, 197–200. https://doi.org/10.1145/2396636.2396667