



HAL
open science

HorizontalDragger: a Freehand Remote Selector for Object Acquisition

Siju Wu, Amine Chellali, Samir Otmane, Guillaume Moreau

► **To cite this version:**

Siju Wu, Amine Chellali, Samir Otmane, Guillaume Moreau. HorizontalDragger: a Freehand Remote Selector for Object Acquisition. IEEE Virtual Reality 2015, Mar 2015, Arles, France. pp.307–308, 10.1109/VR.2015.7223418 . hal-01140514

HAL Id: hal-01140514

<https://hal.science/hal-01140514>

Submitted on 8 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HorizontalDragger: a Freehand Remote Selector for Object Acquisition

Siju Wu*
IBISC, Evry University

Amine Chellali†
IBISC, Evry University

Samir Otmane‡
IBISC, Evry University

Guillaume Moreau§
Ecole Centrale de Nantes - CERMA

ABSTRACT

Interaction with computers using freehand gestures becomes more and more popular. However, it is hard to make precise inputs by making gestures in air without a physical support. In this paper, we present *HorizontalDragger*, a new selection technique aimed to improve the selection precision by converting the 2D selection problem into a 1D selection problem. After setting a region of interest in the display, all the objects inside the region are considered as potential targets. The user can drag the index finger horizontally to choose the desired target.

Index Terms: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques;

1 INTRODUCTION

With the help of sophisticated tracking systems or low-cost depth cameras, it is now possible to make remote interaction with the computer by using bare hands. However, it still lacks a satisfying method to select an object precisely from dense target clusters. In this paper, we present *HorizontalDragger* to simplify object selection from dense target clusters. *HorizontalDragger* is a two-phase selection method which can be performed by bare hand gestures (Figure 1). The main idea of *HorizontalDragger* is to decompose a direct selection task into a coarse 2D selection, followed by a precise 1D selection task, so there are fewer degrees of freedom to control at the same time. Moreover, the effective width of all the potential targets is a constant value so that the selection task is less influenced by the target size and the density of the target distribution.

2 RELATED WORKS

Ray-casting is the most used approach to make remote selection [4]. However, to select small objects from dense target distributions, it requires great concentration to stabilize the cursor. One solution to simplify remote selection is to provide two operation modes to switch CD ratio: *Coarse* mode and *Precise* mode. CD ratio is the ratio between input movements and output movements. The work of [10] [7] and [6] propose different solutions to switch CD ratio.

When performing *Ray-casting*, sometimes more than one object is crossed by the line ray. *Depth ray* and *Lock ray* are two techniques which permit the user to pick up one object among all those crossed by the ray [2]. *3D Bubble Cursor* is another technique which can be used to select objects of different depths [9].

The marking menu is also used to rearrange objects in a spare distribution to facilitate the selection task. *Stroke* and *Reach* are two techniques which allow the user to perform strokes to select an option in the menu [8]. *Flower ray* proposed in [2] can rearrange objects crossed by the ray line in a circular marking menu. SQUAD

[3] uses a similar strategy to regroup several objects in a marking menu for precision refinement.

3 DESIGN AND IMPLEMENTATION

In this section, we first present the design of *HorizontalDragger* and then explain the implementation details.

3.1 Design

HorizontalDragger is designed to allow the user making remote selection through bare hand gestures. This technique is inspired by *LinearDragger* proposed in [1]. We borrowed its idea and extent *LinearDragger* to the 3D space.

The technique principles are as follows: after stretching the index finger of the active hand, a circle is displayed on the screen. The circle follows the movement of the index finger and can be used to determine the region of interest (ROI) (Figure 1 (a)). To set the ROI, the user should perform a thumb clicking gesture by stretching the thumb and then retrieving it (Figure 1 (b)). After performing this gesture, the circle is fixed and the region inside the circle is set as the ROI. All the objects inside the ROI are considered as potential targets. Once the ROI is set, the user can drag the hand horizontally to scan all the potential targets one by one and the target checked by the user is highlighted (Figure 1 (c)). The dragging direction is parallel to the display. The scanning order is determined by checking the horizontal coordinates of all the potential targets. To select the highlighted target, the user needs to make another thumb clicking gesture (Figure 1 (d)). When the ROI is fixed, a magnified view of the ROI is provided to facilitate the observation of the potential targets in the ROI. Inside the magnified view, two segments are drawn for each potential target to connect it to its predecessor and successor in the scanning order. This helps the user predict how the focus changes.

The horizontal distance between the current position of the index finger and its recorded position when setting the ROI is used to determine which potential target is highlighted. This way, the scanning order of the potential targets is mapped uniformly to the horizontal moving distance of the index finger. Selecting a target from the ROI feels like dragging a horizontal scroll bar. Since only one motor space requires to be controlled, the user no longer needs to refine the cursor position precisely.

3.2 Implementation

The mapping function of *HorizontalDragger* uses a constant effective width to switch the focus in the ROI. The mapping function is shown in equation (1).

$$k = \begin{cases} 1 & \text{if } P_x - P_{x0} < 0, \\ (P_x - P_{x0})/EW + 1 & \text{if } P_x - P_{x0} \geq 0 \text{ and } P_x - P_{x0} < N * EW, \\ N & \text{if } P_x - P_{x0} \geq N * EW \end{cases} \quad (1)$$

In the function, k is the index of the focused target, P_x is the horizontal coordinate of the current position of the index finger and P_{x0} is the horizontal coordinate of the starting position of dragging, EW is the effective width and N is the count of potential targets. When *HorizontalDragger* is triggered, the index finger is at the starting position so that the first target in the scanning list is focused and highlighted. The target at the far left in the scanning list is selected

*e-mail: siju.wu@ufrst.univ-evry.fr

†e-mail: amine.chellali@ufrst.univ-evry.fr

‡e-mail: samir.otmane@ufrst.univ-evry.fr

§e-mail: guillaume.moreau@ec-nantes.fr

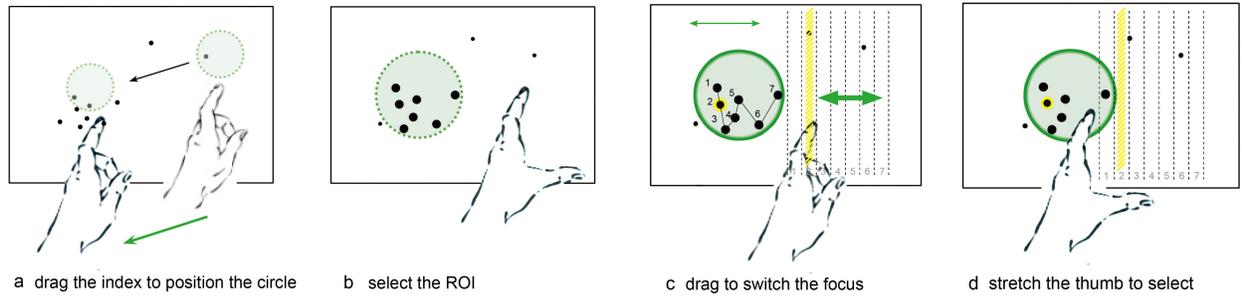


Figure 1: The selection procedures of *HorizontalDragger*

as the first target. The user can drag the finger to the right to switch the focus to the next objects. Dragging to the left allows the user to switch back to previous objects. *EW* is used to control the sensibility of the dragging movement.

4 PRELIMINARY STUDY

We have made two preliminary studies to evaluate the influence of size of the ROI and the effective width. The experiment was conducted on a ThinkPad T510i laptop (Figure 2). The freehand movement is captured using a Leap Motion controller located in front of the laptop with 10cm of distance. The study application was developed using Unity3D with C#.

Eight participants, aged between 25 and 42, were recruited. They are all recruited from a university. None of them have used the Leap Motion controller before and they all happened to be right handed.

Each participant was asked to use *HorizontalDragger* to select the desired target of 12 pixels (3mm) in diameter positioned randomly in the display. 12 candidate targets of the same size were distributed around the desired target. 4 of them were around the desired target with the distance of 18 pixels (4.7mm) and the other 8 with the distance of 36 pixels (9.4mm). Besides, 20 distractor targets of the same size were positioned in the remaining empty space.

In preliminary study 1, each participant was asked to perform the selection task with 5 different ROI diameters: 26, 40, 52, 66 and 80 pixels (6.8, 10.5, 13.6, 17.2 and 20.9mm). For each ROI configuration, each participant performed 20 selection trials. We found that the configuration of 40 pixels had the shortest selection time (3800ms) and the smallest error rate (6.3%).

In preliminary study 2, the procedure was almost the same as study 1. The size of the ROI was set to 40 pixels and each participant was asked to use *HorizontalDragger* with different effective widths: 20mm, 35mm and 50mm. Participants were asked to perform 20 selection tasks for each configuration. The average selection times were 4060ms, 3665ms and 3570ms when the effective width was 20mm, 35mm and 50mm, respectively. The error rates for the different effective widths were 11.1%, 3.3% and 5.2%. We found that the effective width of 35mm and 50mm were better than that of 20mm regarding both selection time and error rate. However, no significant difference was found between 35mm and 50mm.

5 CONCLUSION & PERSPECTIVES

In this paper, we introduced *HorizontalDragger*, a new freehand remote selection technique for small object acquisition. To improve the selection precision, the 2D selection problem was converted into a 1D selection problem. We have made two preliminary studies to find the appropriate size of the ROI and the effective width. In the future work, we will evaluate the performance of our technique by comparing it with other recent remote selection techniques.

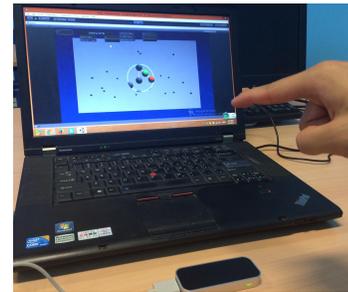


Figure 2: Experimental setup.

REFERENCES

- [1] O. K.-C. Au, X. Su, and R. W. Lau. Lineardrager: a linear selector for target acquisition on touch screens. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2607–2616. ACM, 2014.
- [2] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2006.
- [3] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3D selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 67–74. IEEE, 2011.
- [4] J. Liang and M. Green. Jdcad: A highly interactive 3d modeling system. *Computers & graphics*, 18(4):499–506, 1994.
- [5] M. E. Mott and J. O. Wobbrock. Beating the bubble: using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 733–742. ACM, 2014.
- [6] M. Nancel, O. Chapuis, E. Pietriga, X.-D. Yang, P. P. Irani, and M. Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 831–840. ACM, 2013.
- [7] M. Nancel, E. Pietriga, M. Beaudouin-Lafon, et al. Precision pointing for ultra-high-resolution wall displays. *Technical report*, 2011.
- [8] G. Ren and E. O’Neill. 3d selection with freehand gesture. *Computers & Graphics*, 37(3):101–120, 2013.
- [9] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *3D User Interfaces, 2007. 3DUI’07. IEEE Symposium on*. IEEE, 2007.
- [10] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42. ACM, 2005.