



**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. 9es journées de l'Association Française de Réalité Virtuelle (AFRV), Augmentée, Mixte et d'Interaction 3D, Nov 2014, Reims, France. hal-01101794

**HAL Id: hal-01101794**

**<https://hal.science/hal-01101794v1>**

Submitted on 3 Mar 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

S. Wu<sup>1</sup>, A. Chellali<sup>1</sup>, S. Otmane<sup>1</sup> et G. Moreau<sup>2</sup>

<sup>1</sup>IBISC, Université Evry

<sup>2</sup>LUNAM Université, Ecole Centrale de Nantes - CERMA

---

## Résumé

*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. As a result, selecting a small object from dense target clusters is still a difficult task for bare hand interaction. In this paper, we present the 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. Because the effective width of each potential target is set to a constant value, the user can switch the focus in the region of interest in a predictable way. We have made two preliminary studies to find appropriate values for the size of the region of interest and the effective width. We are going to evaluate the performance of this technique by comparing it with other recent remote selection techniques in the future work.*

---

**Mots clé :** Natural interface, remote selection, gesture

## 1. Introduction

In some interaction scenarios, the user has to stand in front of a large display to interact with a computer. With the help of sophisticated tracking systems or low-cost depth cameras, it is now possible to make remote interaction by using hand-held devices or bare hands. Since people are familiar with using hands to manipulate objects and make gestures in their everyday lives, this kind of interaction mechanism is more intuitive and walk-up-and-use. However, it still lacks a satisfying method to select an object precisely from dense target clusters due to two problems. First, compared to touch based inputs, there is one more motor space to control in order to make gestures in air. Second, freehand interaction is less accurate due to the lack of haptic feedback.

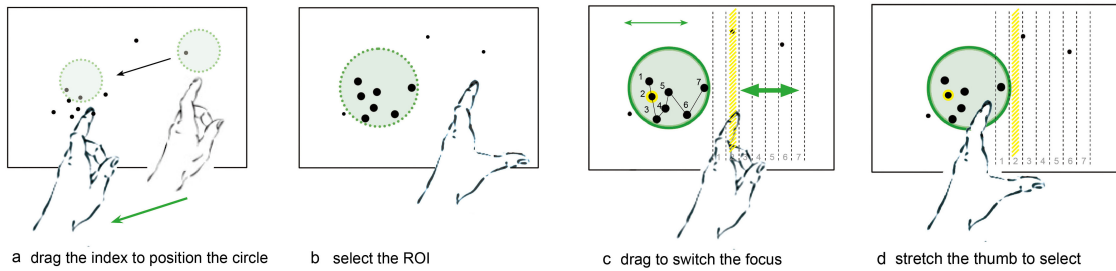
In this paper, we present the new *HorizontalDragger* technique to simplify object selection from dense target clusters. This technique is inspired by the *LinearDragger* proposed in [ASL14]. The *HorizontalDragger* is a two-phase selection method which can be performed by bare hand gestures (Figure 1). When stretching the index finger, a circle is shown in the display. To select an object, the index finger should be moved to make sure the target is covered by the circle. To start the second phase, the thumb of the same hand should be stretched. After that, a region of interest (ROI) is set and all the objects inside it are ordered according to their horizontal coordinates. Then the user only has to drag the index finger

horizontally to switch the focus among all the potential targets. To select the highlighted objects, the thumb should be stretched another time. By decomposing the direct selection task into a coarse 2D selection, followed by a precise 1D selection task, 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

The *Ray-casting* is the most used approach to make remote selection [LG94]. However, to select small objects from dense target distributions, it requires great concentration to stabilize the cursor. To refine selection precision, instead of using a line ray, an aperture or an image plane was used to control the range of the ray [FHZ96] [Pie97]. However, these techniques are neither intuitive nor precise enough to select small size objects.

It is hard to make precise selection using the *Ray-casting* because the Control-Display (CD) ratio is fixed. The CD ratio is the ratio between the input of finger movements and the output of the cursor displacements. One solution to simplify remote selection is to provide two operation modes to switch the CD ratio : *Coarse* mode and *Precise* mode. *Coarse* mode maps hand movements to large scale cursor movements to speed up cursor reposition, while *Precise* mode enables the user to adjust the cursor position finely when get-



**Figure 1:** The selection procedures of the *HorizontalDragger*

ting close to the target. The work of [VB05] allows the user to use the instrumented hand to control the cursor and use different hand gestures to switch the operation mode. Similar selection methods are designed in [NPBL\*11]. Instead of bare hands, it requires a hand-held device to make selection. After using the Smartphone to make coarse pointing, the user can perform the precision refinement on its screen. Another work tried two different ways to manipulate the cursor in *Coarse* mode [NCP\*13]. The cursor can be coarsely moved by the head movement or by dragging two joint fingers on the screen of the Smartphone or the TabletPC. Although different modalities have been attempted to improve the performance of the coarse cursor reposition and precision refinement, the main idea is to switch the CD ratio for different purposes.

When performing the *Ray-casting*, sometimes more than one object is crossed by the line ray. The *Depth ray* and the *Lock ray* are two techniques which permit the user to pick up one object among all those crossed by the ray [GB06]. After casting the line ray, a depth marker can be moved along the line to select the closest object to it. The *3D Bubble Cursor* is another technique which can be used to select objects of different depths [VGC07]. The evaluation in [VGC09] demonstrated that the *3D Bubble Cursor* outperforms the *Depth ray* and the *Lock ray* regarding selection efficiency and accuracy. However, like the *Bubble Cursor*, the performance of the *3D Bubble Cursor* can be degenerated by the distribution density.

The marking menu is also used to rearrange objects in a sparse distribution to facilitate the selection task. The *Stroke* and the *Reach* are two techniques which apply the marking menu to make option selection [RO13]. The user can perform strokes in different directions to select an option in the menu. The *Flower ray* proposed in [GB06] can rearrange objects crossed by the ray line in a circular marking menu. The SQUAD [KBB11] uses a similar strategy to regroup several objects in a marking menu for precision refinement. To avoid losing the focus on the desired target after the rearrangement, the *Expand* [CWL12] is an application of the marking menu which keeps the initial view.

### 3. Design and implementation

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

#### 3.1. Design

The *HorizontalDragger* is designed to allow the user making remote selection through bare hand gestures. This technique is inspired by the *LinearDragger* proposed in [ASL14]. The *LinearDragger* is a touch based selection technique and it is designed to simplify object selection from dense target clusters. We borrowed its idea and extent the *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 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. To make a correct selection, the desired target should be covered by the ROI. Once the ROI is set, the user can drag the hand horizontally to scan 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. Unlike the *LinearDragger* whose scanning order is determined by the direction of the finger motion, the *HorizontalDragger* keeps the horizontal scanning order. Since the performances of freehand dragging along different directions are not the same [RO13], we decided to fix the dragging in the horizontal direction. 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 re-

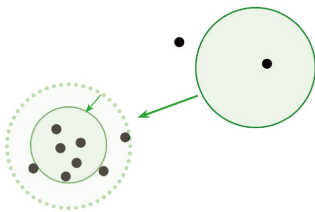
quires to be controlled, the user no longer needs to refine the cursor position precisely.

In general, the *HorizontalDragger* has the following advantages :

- **Freehand interaction** - The *HorizontalDragger* only requires the user to use the bare hand to interact with the remote display. No hand-held device is necessary. Only one hand is necessary to perform the remote selection.
- **Natural gesture** - The gestures used during the selection are familiar to the user. No extra learning is required.
- **Insensitive to target distribution** - The effective width of targets is a constant value. It is not influenced by the distribution density and the target size.
- **Visual guide** - When the magnified view is displayed, auxiliary segments are drawn to help the user understand in which direction the index should be dragged.
- **Adaptable to other system** - Although we have implemented the *HorizontalDragger* by using a Leap Motion controller, it is also possible to implement this selection technique using the Kinect or other sophisticated tracking devices [VB05].

### 3.2. Adaptive radius

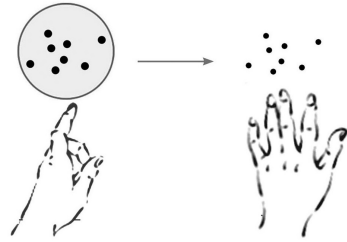
In the first prototype, the radius of the ROI is fixed no matter how many objects it covers. However, after conducting a preliminary study to find an appropriate radius for the ROI, we realized that it was necessary to limit the number of objects inside the ROI. Dragging the finger in air requires more concentration than dragging the finger on the touchscreen. Switching the focus among many potential targets inside the ROI requires longer time and also fatigues the user. To solve this problem, in the second prototype, the radius of the selection circle can be adapted to the density of the cluster inside it (Figure 2). For example, if the maximum number of objects inside the ROI is set to 5, the circle shrinks when there are more than 5 objects inside. The maximum number is set empirically. However, in the two preliminary studies that we are going to introduce, the adaptive radius mechanism is not applied.



**Figure 2:** The adaptive radius. When there are more than 5 objects inside the selection circle, the circle shrinks so that only 5 objects are included.

### 3.3. Selection cancellation

When the desired target is not in the ROI, the user may want to cancel the selection. To avoid wrong selections, we provide one mechanism to cancel the selection. This mechanism requires the user to perform the release gesture by stretching out five fingers (Figure 3). We think that this mechanism is natural because the release gesture is similar to the hand movement made when throwing away something held in hand. In addition, after stretching out the fingers, the hand remains in the same position so that it is easier to prepare for the next selection.



**Figure 3:** The cancellation mechanism. The release gesture is made to dismiss the ROI.

### 3.4. Implementation

The key idea of the *HorizontalDragger* is to use the 1D linear movement of the index finger to switch the focus in the list of potential targets.

The mapping function of the *HorizontalDragger* uses a constant effective width to switch the focus in the ROI. The mapping function is shown in equation (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 the *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 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. We have done a preliminary study to determine its appropriate value.

## 4. Preliminary study 1 : size of the ROI

To find an appropriate size for the ROI, we have conducted a preliminary study to evaluate the performance of the *HorizontalDragger* with different radius of the ROI.

### 4.1. Apparatus

The experiment was conducted on a ThinkPad T510i laptop with a 15'' screen (Figure 4). The resolution of the screen is  $1366 \times 768$ . The freehand movement is captured using a Leap Motion controller located in front of the laptop

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

with 10cm of distance. The study application was developed using Unity3D with C#.

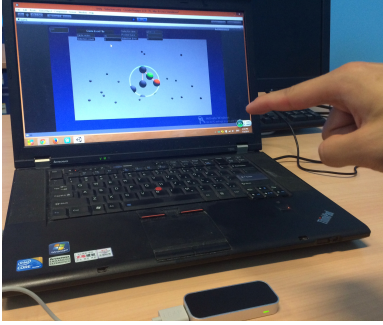


Figure 4: Experimental setup.

## 4.2. Participants

Eight subjects, aged between 25 and 42, were recruited. They are all students and engineers from a university. All of them have experience with touchscreens, but none of them have used the Leap Motion controller before. All the subjects were right handed.

## 4.3. Procedure and Experimental Design

To start the selection task, the user has to drag the cursor and position it on a button located at the left side of the screen. If the cursor remains inside the button for more than 0.5 second, the desired target of 12 pixels (3 mm) in diameter is positioned randomly in a circular region of 100 pixels in diameter. Inspired by the evaluation done in [Vanacken et al. 2009], we display four candidate targets of 12 pixels (3mm) in diameter around the desired target with the distance of 18 pixels (4.7mm). Another 8 candidate targets of the same size are distributed randomly around the desired target with the distance of 36 pixels (9.4mm). Besides, 20 distracter targets of the same size of the candidate targets are positioned in the remaining empty space. Each subject was asked to perform the selection task using the *HorizontalDragger* 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 subject performed 20 selection trials. The average time to accomplish all the trials was about 30 minutes.

The different sizes of ROI are selected according to the target distribution density. With the smallest ROI, nearly only one target can be covered by the circle. When using the largest ROI, almost all the candidate targets around the desired target can be covered by the ROI. We wanted to investigate how the number of potential targets inside the ROI

influences the performance. In this preliminary study, the radius of the ROI was fixed and could not adapt to the object distribution automatically.

During the study and for each selection task, we recorded the task completion time which starts from the moment the user taps the start button and ends when the desired target is selected. If the desired target is not covered by the ROI or a wrong target is selected by the subject, the trial continues until the correct one is selected. Moreover, the error rate was calculated by dividing the count of imperfect tasks by the total count of all the tasks. Once one or more errors are made, the task is considered imperfect.

Before the study begun, we first explained how the *HorizontalDragger* works and the study procedure. Each subject had 5 to 10 minutes to try the *HorizontalDragger* to warm up. Each subject performed a total of 100 trials (20 for each ROI diameter) and the total number of all the selection trials was  $8 \times 5 \times 20 = 800$  trials.

## 4.4. Results & Discussion

The average selection times were 4015ms, 3800ms, 4118ms, 4402ms and 4532ms when the size of the ROI was 26, 40, 52, 66 and 80 pixels (6.8, 10.5, 13.6, 17.2 and 20.9mm) in diameter, respectively. The error rates for the different sizes of the ROI were 7.5%, 6.3%, 7.5%, 10% and 8.2%, respectively. An ANOVA has shown a significant effect of the size of the ROI on the mean selection time ( $F = 3.44, p = 0.008$ ). The Post-hoc tests have shown only a significant difference between 40 and 80 pixels (10.5 and 20.9mm). No significant difference exists among the error rates of different sizes.

We found that when the size of the ROI is 40 pixels (10.5mm), both the performance time and the error rates are the best compared with the other conditions. When the ROI is smaller, fewer objects are covered by it. However, to include the desired target, the subject should position the ROI very close to the desired target. This step takes longer time and sometimes the target is missed. When the size of the ROI is bigger than 40 pixels (10.5mm), it is easier to cover the desired target, but the more objects are included, the longer the time it takes to switch the focus.

After discussing with subjects, they all thought that switching the focus among more than 5 or 6 targets was more tiring and they felt more visually distracted. Hence, instead of fixing the radius of the ROI, we decided to allow them to adapt the size of the ROI to the target distribution. We set the maximum size of the ROI to 40 pixels (10.5mm) according to the study results. The ROI radius shrinks when there are more than 5 objects in it. The mechanism of the adaptive radius was implemented and tested in the main experiment.

## 5. Preliminary study 2 : effective width

Besides the size of the ROI, we have also conducted a study to investigate the influence of the effective width. In this study, the diameter of the ROI was set to 40 pixels (10.5mm). The size of the ROI was fixed and could not adapt to the distribution automatically in this study.

### 5.1. Apparatus and Participants

Same as those used in the preliminary study 1.

### 5.2. Procedure and Design

The procedure in the study 2 is almost the same as the study 1. However, we fixed the size of the ROI, and subjects were asked to use the *HorizontalDragger* to make object selection with different effective widths. Three effective widths were compared in this study : 20mm, 35mm and 50mm. The shorter the effective width is, the quicker to switch the focus in the ROI. However, setting the effective width too small makes the focus switching too sensitive to the dragging movement. Using a larger effective width can make it more stable with the cost of longer dragging time. Hence, we wanted to find an appropriate value of the effective width to keep the balance between the stability and the speed. For each effective width, subjects were asked to perform 20 selection trials so that the total number of all the trials was  $8 \times 3 \times 20 = 480$  trials.

### 5.3. Results & Discussion

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%, respectively. An ANOVA has shown a significant effect of the effective width on the selection time ( $F = 4.907$ ,  $p = 0.008$ ) and also on the error rate ( $F = 5.167$ ,  $p = 0.01$ ). The post-hoc tests have shown significant differences between 20mm and 35mm and also between 20mm and 50mm for both the selection time and the error rate. No significant differences were found between the 35mm and 50mm effective widths for the selection time and for the error rate. The study results show that the performances of the second and the third effective widths are better than the first one regarding both the selection time and the error rates. Most of the subjects thought that the shortest effective width is too sensitive to tiny dragging movement. To make a correct selection, it requires more concentration to drag the finger carefully in order to switch the focus to the desired target. Although 6 people thought that the effective width of 35mm is enough and 50mm is a little too long, the results showed that the third effective width takes a little less time. Hence, to stabilize the performance, we chose the 50mm for the main evaluation study.

## 6. Conclusion & Perspectives

In this paper, we introduced the *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 want to improve the performance of the *HorizontalDragger*. For example, inspired by [MW14], we hope the magnified view of the ROI can be opened automatically when the finger approaches the target. We will evaluate the performance of our technique by comparing it with other recent remote selection techniques.

## Références

- [ASL14] AU O. K.-C., SU X., LAU R. W. : Linear-dragger : a linear selector for target acquisition on touch screens. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), ACM, pp. 2607–2616.
- [CWL12] CASHION J., WINGRAVE C., LAVIOLA J. J. : Dense and dynamic 3d selection for game-based virtual environments. *Visualization and Computer Graphics, IEEE Transactions on. Vol. 18*, Num. 4 (2012), 634–642.
- [FHZ96] FORSBERG A., HERNDON K., ZELEZNIK R. : Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology* (1996), ACM, pp. 95–96.
- [GB06] GROSSMAN T., BALAKRISHNAN R. : 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* (2006), ACM, pp. 3–12.
- [KBB11] KOPPER R., BACIM F., BOWMAN D. A. : Rapid and accurate 3d selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on* (2011), IEEE, pp. 67–74.
- [LG94] LIANG J., GREEN M. : Jdcad : A highly interactive 3d modeling system. *Computers & graphics. Vol. 18*, Num. 4 (1994), 499–506.
- [MW14] MOTT M. E., WOBROCK J. O. : 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* (2014), ACM, pp. 733–742.
- [NCP\*13] NANCEL M., CHAPUIS O., PIETRIGA E., YANG X.-D., IRANI P. P., BEAUDOUIN-LAFON M. : High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), ACM, pp. 831–840.
- [NPBL\*11] NANCEL M., PIETRIGA E., BEAUDOUIN-LAFON M., ET AL. : Precision pointing for ultra-high-resolution wall displays. *Technical report* (2011).
- [Pie97] PIERCE, JEFFREY S AND FORSBERG, ANDREW S AND CONWAY, MATTHEW J AND HONG, SEUNG AND ZELEZNIK, ROBERT C AND MINE, MARK R : Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics* (1997), ACM, pp. 39–ff.

- [RO13] REN G., O'NEILL E. : 3d selection with freehand gesture. *Computers & Graphics*. Vol. 37, Num. 3 (2013), 101–120.
- [VB05] VOGEL D., BALAKRISHNAN R. : Distant free-hand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), ACM, pp. 33–42.
- [VGC07] VANACKEN L., GROSSMAN T., CONINX K. : 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* (2007), IEEE.
- [VGC09] VANACKEN L., GROSSMAN T., CONINX K. : Multimodal selection techniques for dense and occluded 3d virtual environments. *International Journal of Human-Computer Studies*. Vol. 67, Num. 3 (2009), 237–255.