Multi-Touch Smartphone-Based Progressive Refinement VR Selection

Elaheh Samimi*
Carleton University, Ottawa, Canada

Robert J. Teather[†]
Carleton University, Ottawa, Canada

Abstract

We developed a progressive refinement technique for VR object selection using a smartphone as a controller. Our technique combines progressive refinement with the marking menu-based CountMarks, which uses multi-finger touch gestures to "short-circuit" multi-item marking menus. Users can indicate a specific item in a sub-menu by pressing a specific number of fingers on the screen while swiping in the desired menu's direction. This reduces the number of steps in progressive refinement selection.

Keywords: Selection, progressive refinement, touchscreen.

Index Terms: Human-centered computing → Virtual reality

1 Introduction

Progressive refinement is a relatively recent class of selection techniques introduced by Kopper et al. [3], which break a selection task down into sequence of smaller selections. Users do not need to select the target object precisely in the first step. Instead, they select a cluster of objects near the target without high precision. A 2D quad menu appears, containing all objects selected in the first cluster. The user then selects the quadrant containing the target object, refining the selection, and repeats until just the target object remains. Such techniques provide high precision in exchange for selection speed, and work well for dense virtual environments.

The CountMarks technique [6] offers a potential improvement for progressive refinement. CountMarks is a menu-based selection technique for smart-glasses or AR HMDs. It uses a smartphone as a controller; previous work has shown that smartphones show promise as VR controllers [1, 4], with recent VR devices, like the HTC Vive Flow[1] using smartphones as a primary controller.

CountMarks combined count menus [2], where the user touches a number of fingers to the screen corresponding to the desired item of selection, with marking menus [5], a gesture-based method where a pie menu allows users to select their desired items by swiping their finger in the correct direction (e.g., up, down, right, left). With CountMarks, multiple items were shown on each sub-menu of a radial menu. The number of fingers touched to the screen indicates which item in a menu should be selected, while swiping in a direction indicates which menu should be selected. CountMarks reduces the number of menus open, improving selection time.

CountMarks' directional menu makes it complimentary with progressive refinement. Based on this observation, we developed a new progressive refinement technique employing CountMarks to "short-circuit" the menu hierarchy in progressive refinement by skipping some menu levels for selecting the target object inside the quad menu. Our technique supports "careless" selection during the first step, like other progressive refinement techniques by using sphere-casting and distributing objects into quad menus. Multi-touch finger input supports picking between multiple objects in a quad menu, reducing the number of refinement steps required relative to previous progressive refinement techniques.

2 Technique Design

We designed a multi-touch progressive refinement technique to improve selection speed while offering high accuracy. Our technique was implemented in Unity, using an Oculus Rift CV1 head-mounted displays, and Samsung Galaxy S8 smartphone as the input device. Our technique requires several steps to accomplish selection. We describe each step below:

Step 1: Select Target Area: Like SQUAD and other progressive refinement techniques, we employ ray-casting with a selection sphere at the ray/scene intersection point. Objects in the sphere are selection candidates. See Figure 1.

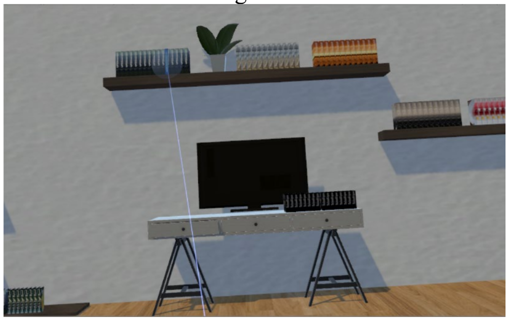

Figure 1. First step, selecting target area via sphere-casting.

Step 2: Displaying Selected Objects: The initial selection typically yields multiple candidate objects. We spread candidate objects across a quad menu, similar to SQUAD. The objects are positioned into predefined spatial positions on the quad menus, and distributed evenly inside each menu quadrant. The number and size of objects in each quadrant depends on environment density. More densely packed virtual environments will tend to have more objects in the initial selection than low-density environments, and objects scale to fit quadrants. See Figure 2.

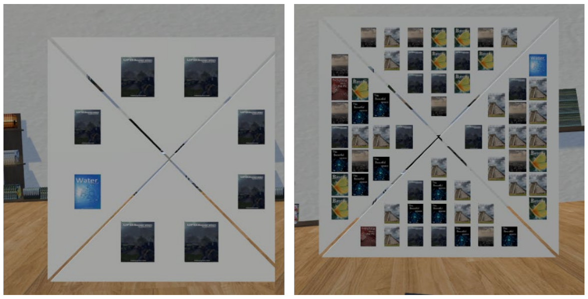

Figure 2. Second step, displaying selected objects. Left) distributed objects inside a quad menu in low density. Right) distributed objects inside a quad menu in high density

[1] https://www.vive.com/us/product/vive-flow/overview/

*email: elahehsamimi@cmail.carleton.ca
[†] email: rob.teather@carleton.ca

To spread objects across quadrants, we first fill the first position in each, then each second position, and so on. See Figure 3.

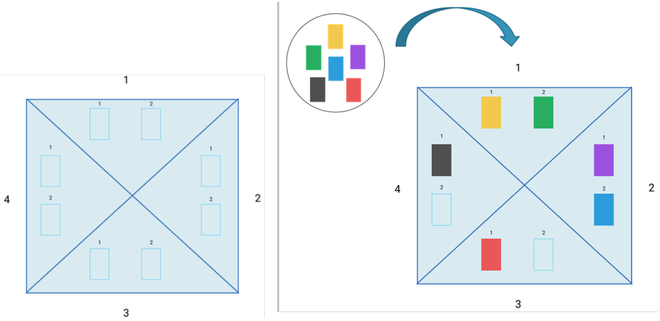

Figure 3. Ordering objects inside the quad menus. Left) Predefined spatial positions in each quadrant. Right) distributing candidates.

Finally, we display the quad menu. If there are fewer than four items in each quad menu, we additionally present a number for each object on each quad menu. This corresponds to how many fingers would select that object with a swipe. See Figure 4.

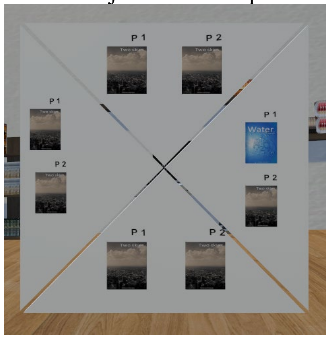

Figure 4. Quad menu, depicting object positions, i.e., number of fingers to select a given item. For example, the blue book is in position 1; touching one finger and swiping right would select it.

Each menu can support 16 objects (maximum number of menu items in marking menu technique was eight) according to the size of each quadrant menu, size of each object inside the quadrant menu, and spaces between them. Hence, we can support up to 16 (objects) × 4 (number of menus) = 64 objects in each menu level.

Step 3: Selecting an Object: If the desired object was in the initial selection, the user next refines the group to remove undesired objects. Up to this point, our technique is similar to others [3]. This step is the main difference of our technique:

Step 3.1 Indicating the target via multitouch: This step employs the idea of CountMarks [6]. If there are four or fewer objects in the menu, the user can select it outright by touching the smartphone screen with the appropriate number of fingers corresponding to the target's position. The technique currently supports up to four fingers. As the user touches the screen, the object corresponding to the finger count is highlighted yellow. For instance, in Figure 5, the user has put one finger on the screen so as to select the target object (i.e., the blue book), which is in the first position. With one finger on the screen, all objects in the first position in each quadrant are highlighted to yellow as feedback.

***Step 3.2 Select the Target Object by Swipe Gesture*:** After indicating the desired object by touching the screen (as described above), users finalize selection by swiping their fingers in the desired direction. As soon as the user moves their fingers in the desired direction, the selected quadrant is highlighted green for feedback. For example, in Figure 5, the user is swiping their finger left, highlighting the left quadrant green. The selection is completed when the user swipes their finger to the edge of the touchscreen through the desired menu quadrant.

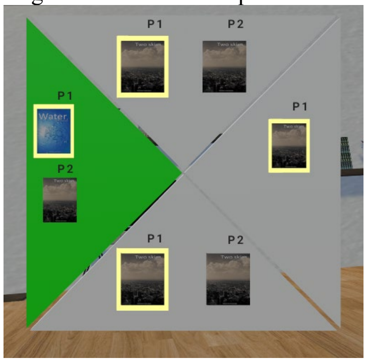

Figure 5. Object highlight: one finger is touching the screen, highlighting all "first" position items. Swipe selection: a left swipe selects the chosen item in the left menu.

The technique currently supports only the four primary swiping directions. However, this could be extended to support more (e.g., up to eight) directions and menus, and to display more selected items, or to distribute items further across each menu. In situations where there are more than four candidate objects in a quadrant (i.e., more objects than the user has fingers), users first refine the selection further before multi-touch selection is possible. This is seen, for example, in Figure 2 (right). The user refines by swiping any number of fingers towards the desired quad menu. The menu items from the selected quad redistribute, and the process repeats from Step 2, until there are four or fewer items on each quad. At this point, they can then select the object as described in Step 3.

3 Conclusion

We developed a VR selection technique that combines progressive refinement with the CountMarks multi-touch marking menu technique. While previous progressive refinement techniques are known to improve accuracy, our technique helps with speed by using multi-touch to reduce the required menu navigation. Future work will validate our approach through an empirical study of our technique, compared to other progressive refinement methods such as SQUAD.

References

[1] Babic, T., Reiterer, H., and Haller, M., Pocket6: A 6DOF controller based on a simple smartphone application, *ACM Symposium on Spatial User Interaction - SUI* 2018, 2-10.

[2] Bailly, G., Lecolinet, E., and Guiard, Y., Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces, *ACM Conference on Human Factors in Computing Systems - CHI* 2010, 591-594.

[3] Kopper, R., Bacim, F., and Bowman, D. A., Rapid and accurate 3D selection by progressive refinement, *IEEE Symposium on 3D User Interfaces - 3DUI 2011*, 67-74.

[4] Kyian, S. and Teather, R.J., Selection performance using a smartphone in VR with redirected input, *ACM Symposium on Spatial User Interaction - SUI* 2021, 1-12.

[5] Lepinski, G. J., Grossman, T., and Fitzmaurice, G., The design and evaluation of multitouch marking menus, *ACM Conference on Human Factors in Computing Systems - CHI* 2010, 2233-2242.

[6] Pollock, J. and Teather, R. J., CountMarks: Multi-finger marking menus for mobile interaction with head-mounted displays, *International Conference on Human-Computer Interaction* - HCII 2020, 241-260.