Rapid and accurate 3D selection by progressive refinement

Regis Kopper* Felipe Bacim† Doug A. Bowman‡

Dept. of Computer Science and Center for Human-Computer Interaction
Virginia Tech

Abstract

Issues such as hand and tracker jitter negatively affect user performance with the ray-casting selection technique in 3D environments. This makes it difficult for users to perform tasks that require them to select objects that have a small visible area, since small targets require high levels of precision. We introduce an approach to address this issue that uses progressive refinement of the set of selectable objects to reduce the required precision of the task. We present a design space of progressive refinement techniques and an exemplar technique called Sphere-casting refined by QUAD-menu (SQUAD). We explore the tradeoffs between progressive refinement and immediate selection techniques in an evaluation comparing SQUAD to ray-casting. Both an analytical evaluation based on a distal pointing model and an empirical evaluation demonstrate that progressive refinement selection can be better than immediate selection. SQUAD was much more accurate than ray-casting, and SQUAD was faster than ray-casting with small targets and less cluttered environments.

Keywords: 3D interaction, 3D selection, progressive refinement, distal pointing.

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

1 Introduction

Selection, which involves the specification of one or more objects by the user, is one of the fundamental tasks in 3D environments [1]. Although various metaphors for selection of single objects have been developed, such as virtual-hand [19] and image-plane techniques [15], ray-casting [13] is perhaps the most popular selection style in VEs due to its simplicity and generality. Ray-casting requires only two degrees of freedom and works at any distance, while virtual hand techniques require at least three degrees of freedom and are often limited to a certain distance from the user. However, even though ray-casting provides better performance than virtual hand techniques in many applications, it has limitations. When the visual size of the target is small, due to the object size, occlusion, or distance from the user, ray-casting is slow and error-prone [17], because it does not provide high-precision pointing at a distance.

The first IEEE 3DUI Grand Prize [2] provides an example of a selection task for which ray-casting is unsuitable. The contest environment was a virtual supermarket where, among other tasks, selection of occluded objects in a highly cluttered environment was required. To simply use ray-casting for this task would require the user to select partially occluded targets very precisely from a distance, to remove occluding objects to increase the visual size of the targets, or to spend time traveling close to the targets to make them easier to select. Any of these options would have a high error probability and/or take a long time to achieve.

A number of techniques have been proposed to deal with the precision limitations of ray-casting. Examples include snapping [7], cone-casting [10], 3D bubble cursor [20] and PRISM-style pointing [4]. However, some of these techniques do not perform well in highly cluttered environments. They require users to interact very carefully to accomplish a single precise selection, and may actually provide worse performance than standard ray-casting in some situations. There are several tasks that could benefit from techniques that allow accurate selection in cluttered environments without requiring users to be precise. Examples include tasks that involve interaction with very large data sets, such as astrophysical or atomic datasets, in addition to the supermarket task already mentioned.

To address this challenge, we propose a selection method that uses progressive refinement of the set of selectable objects. The main idea is to gradually reduce the set of selectable objects until the target is the only one left without requiring the user to be precise at any point during the refinement. This is in contrast to traditional techniques that use immediate selection, requiring precision. We can see an inherent tradeoff between these two categories of techniques. Progressive refinement requires a process of selection, often using multiple steps, although each step can be very fast and accurate. Immediate techniques, on the other hand, involve a single high-precision spatial selection at the expense of being slow and having a higher error probability. The goal of the work presented in this paper is to explore this tradeoff to determine whether progressive refinement techniques are useful. In other words, we want to know when it makes sense to sacrifice the simplicity of immediate selection in order to improve speed and accuracy.

As an example of selection by progressive refinement, we designed the Sphere-casting refined by QUAD-menu (SQUAD) selection technique, which uses two distinct refinement phases. In the first phase, the user specifies a volume containing the target object. The user then refines the initial selection progressively by selecting the subset of objects containing the target from a four-item menu displaying all the remaining objects, until the target is finally selected. SQUAD makes it possible to accomplish precise selection without requiring the user to use precise actions at any moment during the selection task.

We hypothesize that SQUAD and other progressive refinement selection techniques have nearly perfect accuracy. This is because these techniques can successfully select any object, no matter how cluttered the scene, or how small the object, by allowing users to make refinements in an imprecise, careless manner. We also believe that selection techniques based on progressive refinement can be faster than immediate techniques in cases where targets are small, as long as the number of required refinements is not too high. To test these hypotheses, we compared SQUAD to standard ray-casting analytically (using a predictive model of distal pointing [9] and a novel progressive refinement selection model), and empirically (through a controlled user study).

2 Related Work

Bowman *et al.* [1] divided selection techniques into four main categories: selection by pointing (e.g., ray-casting [13]), selection

*e-mail: kopper@vt.edu
†e-mail:fbacim@vt.edu
‡e-mail:bowman@vt.edu

by touching (e.g., virtual-hand [14]), selection by occlusion (e.g., image-plane techniques [15]) and indirect selection (e.g., selection by attributes [1]). Most of these techniques can be classified as immediate selection, since they only require a single high-precision selection without refinement.

Ray-casting [13] is a widely used pointing-based technique, in which the user points with a virtual ray extending from the hand or input device to specify an object in the scene. Although it is very simple, this technique in its pure form suffers from a number of issues, mostly because of natural hand tremor and tracker jitter, which make it difficult for the user to control the origin and orientation of the ray. This is a bigger issue with ray-casting than with other techniques because the small hand movements are amplified at the end of long rays, causing ray-casting to be less precise as target objects get farther from the user. These issues make ray-casting difficult to use when the objects have a small visual size [17], as selecting such objects by pointing requires high levels of precision.

In order to address these issues, a number of improvements have been proposed. Even though such techniques improve selection performance in general, they can have a negative effect in very cluttered environments. Cone-casting [10], for example, extends ray-casting by adding a cone-shaped volume to the ray to make it easier to select objects that are distant. In cluttered environments, however, many objects will fall inside the cone, so that the user still has to point precisely to select the desired object. The snapping technique presented by Haan *et al.* [7] uses a selection volume to calculate and accumulate scores over time for each object. This way, it can estimate which object the user wants. The bubble-cursor [5] is a 2D technique that dynamically resizes a circular cursor so that it only contains one object. A 3D extension of the bubble-cursor, which uses a sphere instead of a circle, was presented by Vanacken *et al.* [20]. Both of these techniques may actually perform worse in cluttered environments, since even small movements will cause the ray or the cursor to constantly snap or resize to select new targets.

Other techniques improve ray-casting accuracy by changing the control-display ratio, either automatically (e.g., PRISM [4], when moving slowly) or manually (e.g., ARM [9], by pressing a button), or by providing the ability to zoom (e.g., zoom-and-pick [3]). While these techniques can achieve very high levels of precision, all of them have limitations. PRISM and ARM cause a significant mismatch of the physical pointing direction to the perceived pointing position, and the mapping is nonlinear. Zoom-based techniques suffer from potential loss of detail. Finally, these techniques require the user to interact very carefully and with full attention. Our proposed approach of selection by progressive refinement aims to allow "lazy" interaction with high accuracy.

There are existing progressive refinement techniques in the literature. For example, the shadow cone-casting technique [18] uses continuous movement with cone-casting to disambiguate selection. In [16], Steed presented a general model for selection using 3D gestures and proposed a range of techniques that can use the same concepts. The depth-ray technique [6], which adds depth control to the classic ray-casting technique to select occluded objects, requires two actions to specify the target. PORT [11] allows the selection of multiple objects and uses a series of movement and resizing actions to define the set of targets.

The flower ray technique [6], in which occluding targets are concurrently selected by ray-casting and disambiguated in a second phase by a marking menu is perhaps the closest existing technique to our approach. Although this technique is suited for highly cluttered environments, it requires high precision for the ray selection and does not scale well, since the marking menu object specification is done in a single phase.

To the best of our knowledge, there has been no prior generalization of the progressive refinement concept, and no comparison of progressive refinement techniques to immediate techniques.

3 Selection by Progressive Refinement

As we stated above, the concept of selection by progressive refinement is to gradually reduce the set of selectable objects until only the target remains. We identify three dimensions to the design space, which are shown in Figure 1.

- Type of progression
  - discrete
  - continuous
- Refinement criteria
  - spatial
  - by object attributes
  - out-of-context
- Display of selectable objects
  - in context
  - out-of-context

Figure 1: Design space of selection by progressive refinement.

First, progressive refinement can be done either through several discrete steps, as in SQUAD, or with a continuous process, as in shadow cone-casting [18].

The method of refinement defines another dimension of the design space. This refers to the criteria that are used to reduce the set of selectable objects. Refinement can be specified spatially within the environment context, for example through the use of a volume or area in the image plane, limiting the region of the environment where the target can be. Refinement can also be by the specification of attributes of the desired object, such as color, size or shape. Refinement can also be done through "out-of-context" subset specification, which involves picking a subset of objects from a list or menu instead of from the environment. The quad-menu refinement in SQUAD is an example.

The design space is further defined by the method used to display the current set of selectable objects. Subsets of selectable objects can be displayed in context, for example through zooming, visual explosion, highlighting, moving the viewpoint closer to the subset or through the removal or dimming of non-selectable objects. The subset of selectable objects can also be displayed out-of-context, through the use of menus, which may be sorted in some way or arranged randomly.

We can also characterize progressive refinement selection techniques along a continuum based on the gradualness of refinement. At one end of the spectrum we have the immediate techniques, which directly specify the target object. This can be thought of as a "refinement" from the entire set of selectable objects in the environment to one or zero (in case of a failed selection) in a single step. At this end of the continuum, too much precision may be required, as an exact element needs to be specified immediately. At the other end of the continuum we can imagine a technique that has many refinement steps, with an extreme case being a technique where each refinement simply excludes one object from the set of selectable objects. Here precision is also required, and in fact such a technique requires many high-precision selections. In the middle of the continuum are the techniques of interest, where the reduction in the set of selectable objects is rapid and accurate.

3.1 SQUAD Selection

We designed a progressive refinement selection technique that falls in the middle of the gradualness continuum. It uses discrete progression, a combination of spatial and out-of-context refinement methods, and a combination of in-context and out-of-context display of the subset of selectable objects. We call this technique Sphere-casting refined by quad-menu (SQUAD).

We designed SQUAD as part of our entry to the 3DUI Grand Prize contest, described in [2]. The main challenge proposed by the contest was to design techniques that support interaction in a highly cluttered environment. In a virtual supermarket, users had to

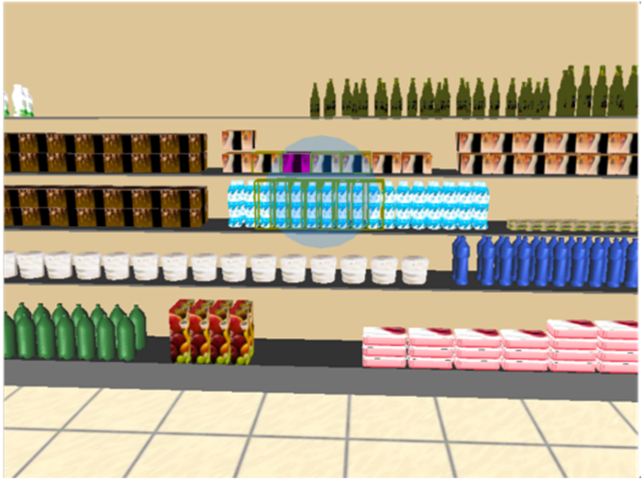

Figure 2: Sphere-casting.

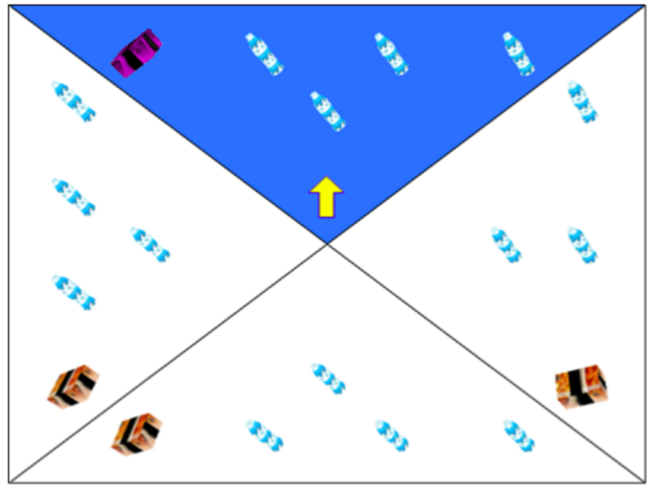

Figure 3: Quad-menu. Note that the target object needs to be visually distinct for the selection to be feasible.

select specific objects identified by textures with unique characteristics. To achieve rapid yet precise selection, we designed SQUAD as a progressive refinement technique that divides selection into two discrete steps, the first being spatial and in-context and the second being out-of-context.

The first step uses a modified version of ray-casting that casts a sphere onto the nearest intersecting surface to determine which objects will be selectable. We call this subtask sphere-casting. The user simply has to ensure that the desired object is inside or touching the sphere, so that it can be picked from among the other objects in the next phase. Items that will be made selectable are highlighted. In order to improve confidence that the desired object will be available, the sphere's radius increases the farther the user is from the nearest intersecting surface, thus increasing the overall number of objects available in the second phase. Figure 2 illustrates this selection phase. (Note that in the study described in section 4, however, the sphere size is fixed since all objects are placed at the same distance from the user.) Sphere-casting avoids the precision issues of ray-casting, and also allows selection of occluded objects.

Upon completion of the first phase, all objects that were inside or touching the sphere are evenly distributed among four quadrants on the screen, without regard for the spatial locations of the objects in the 3D environment. We call this the quad-menu, and note its similarity to marking menus. Contrary to zone menus [23], where breadth of selection is achieved by relative position of multiple marking gestures, in the quad menu phase users refine the selection by repeatedly pointing anywhere in the quadrant that contains the item they are looking for, each time reducing the number of objects per quadrant until the desired object is the only one left. This process is illustrated in Figure 3. The maximum number of selections necessary in the quad menu is $\lceil \log_4 n \rceil$, where n is the initial number of items. For example, if the sphere has between 17 and 64 objects inside it, our technique would require at most four clicks to select the target (one click for sphere-casting and three clicks for the quad-menu).

SQUAD is an example of a progressive refinement technique that works well in environments where there are many objects that are arranged along a surface, and where the desired object is visually distinct from the rest. For other selection tasks or environments, however, different design choices (e.g., using a cone as the selection volume or distributing items in the menu based on spatial location) might be preferred.

4 Evaluation

We conducted a formal evaluation comparing SQUAD to standard ray-casting.

4.1 Experiment Overview

We evaluated the task of pointing at circular targets that varied in radius, on a screen that was filled with distractor objects varying in number and density.

4.1.1 Goals and Hypotheses

The overall goal of the experiment was to explore the tradeoff between ray-casting and SQUAD. While ray-casting requires only one click, it requires precision with visually small targets. SQUAD, on the other hand, requires very little precision from the user, at the expense of multiple steps until the desired target is selected.

With this tradeoff in mind, we expected there to be an interaction between technique and target size. We hypothesized that SQUAD would take constant time with respect to target size, while ray-casting would be slow with small targets and fast with large targets. We were unsure how the constant SQUAD times would compare to the times for ray-casting with the various target sizes, but expected that SQUAD would be faster in at least some target size conditions.

We also hypothesized that the number of distractor objects around the target would have a significant effect on time to select with SQUAD, but that the number of distractors would have no effect on ray-casting. We expected that SQUAD would outperform ray-casting when the number of distractors was small.

With respect to accuracy, we hypothesized that SQUAD would yield virtually no errors, due to its low required precision, whereas ray-casting would have more errors as the target sizes decrease.

Finally, we hypothesized that situations in which the tracking has more jitter would result in more errors and slower time for ray-casting, but would not impact SQUAD, as all the steps of the technique require very low pointing precision.

4.1.2 Design

We used a factorial within-subject design with repeated measures. There were four independent variables: *technique* (ray-casting, SQUAD), *tracking* (normal, jittery), *target size* (radii $0.01m$ or 0.26°, $0.015m$ or 0.40°, $0.04m$ or 1.06°), and the number of distractors inside the selection sphere (referred to as distractor density) (16, 64, 256). After Kopper *et al.* [9], we emphasized in varying the target size, while keeping the movement amplitude within a roughly constant range (see section 4.3.1). The design was, thus, 2*x*2*x*3*x*3.

The order of presentation of technique and tracking was counterbalanced, blocked by technique, such that each participant performed both tracking conditions within the same technique before moving to the next one. Within the combinations of technique and tracking, each of the nine conditions of target size *vs.* distractor density was repeated eight times and presented in random order.

4.2 Analytic Evaluation

Before running an empirical study (section 4.3), we analytically evaluated performance in our experimental conditions based on predictive models.

The tradeoff between speed and accuracy described in Fitts' law is well known for pointing tasks [12, 22]. Recently, a similar model was shown to apply for distal pointing tasks, in which the input device is remotely located with respect to the display area and the pointing is done in a direct fashion, as opposed to indirectly, for example, through the use of a mouse [9]. SQUAD and ray-casting both use distal pointing, making this model relevant to our study.

Kopper *et al.*'s predictive model of distal pointing states that the time to acquire a distal target through direct pointing depends strongly on the angular width of the target and, to a much lesser degree, on the angular amplitude of the wrist/arm movement required to complete the task. The difficulty of the task is expressed as

$$ID_{DP} = \left[\log_2\left(\frac{\alpha}{\omega^k}+1\right)\right]^2, \tag{1}$$

where ID_{DP} is the index of difficulty, α is the angular amplitude of the movement and ω is the angular width of the target. The constant k is a power factor greater than one that expresses the higher importance of the target width relative to movement amplitude. The value of k was shown to be around three in the experimental setting used by Kopper *et al.* While our study used a different environment, we believe that it was similar and the value of the constant k should be approximately the same.

The goal of the progressive refinement technique that we propose is to reduce the index of difficulty of an individual pointing action to a minimum at the expense of increasing the number of actions needed to achieve the goal of selecting a single unique object in a highly cluttered environment.

In order to reduce ID_{DP} to a minimum in our study, we set the diameter of the selection sphere to 26.3°. The targets were chosen within a constant range from the starting point, so that the movement amplitude was selected randomly between 10.0° and 17.9°, with an average α of 14.0°. This yields an ID_{DP} of

$$ID_{DP} = \left[\log_2\left(\frac{14.0}{26.3^3}+1\right)\right]^2 \approx 1.23\times 10^{-6}. \tag{2}$$

Thus, the index of difficulty of the task of selecting the target region becomes virtually zero, and the expected time to select the target is very small. Similarly, the difficulty of selecting a quadrant in the quad-menu is minimal, as the angular width of each of the quadrants is 45°, yielding an ID_{DP} very near zero. According to Kopper *et al.*'s model, the intercept of the regression line for predicted selection times (when ID_DP tends to zero) is $1.091s$. However, values of ID_{DP} this close to zero have not been tested experimentally. With ω higher than α, we anecdotally observed that selection time is typically under the lower limit of $1s$ set in Kopper *et al.*'s model.

During the quad-menu phase of selection, the user needs to first find the quadrant containing the intended target, then point and click to select it. Although the target stands out and is easily distinguishable from the distractors, in theory the time for visual search will be greater when the number of distractors is larger. We base this assumption on the facts that the visual size of the target is smaller in the quad-menu when there are a lot of distractors, and that the perceived contrast diminishes as the objects become smaller [8]. Thus, we hypothesize that the time it takes to select a target using SQUAD selection is

$$MT_{SQUAD} = c + \sum_{i=1}^{N}(c+v_i), \tag{3}$$

expressed in seconds, where N is the number of refinement iterations required during the quad-menu phase of the technique, c is an empirically determined constant related to the time it takes to point at a target whose difficulty tends to zero, and v_i is the visual search time to find the target in the quad-menu before movement starts. We expect v_1 to take the longest time, because, first, there is a switch in interaction mode, from sphere-casting to quad-menu selection, and a change in the visual environment. Also, the number of distractors is at its maximum, and it decreases as refinements are made, reducing the target search space and time. Due to the visual distinction of the target in relation to the distractors in our experimental setting, we expect v_i to be low in all phases of refinement and to not affect selection time by a large amount.

Here, we note some interesting characteristics of SQUAD selection as compared to ray-casting. First, the target size plays no role in the time it takes to complete a selection. We acknowledge that there may be a longer search time for visual segmentation in highly dense environments with small and occluded targets, but the motor movement time is constant once the target has been found. Second, the time it takes to select a target with SQUAD selection is directly proportional to the amount of clutter – or the number of distractor objects that exist in the region of the desired target. While the time it takes to select a target grows linearly with the increased number of iterations, the growth in the number of iterations is rather slow, on the order of $\lceil log_4(n)\rceil$, where n is the number of objects inside the sphere [2].

In order to compare ray-casting with SQUAD, we decided to vary both the target size and the number of distractor objects that fall inside the selection sphere at any given time. We defined target ωs as 0.53°, 0.80° and 2.12°, yielding for ray-casting an ID_{DP} of 42.9, 23.4 and 1.67, respectively. The predicted time to complete the ray-casting tasks for each of the respective target sizes was 2.29s, 1.74s and 1.14s, respectively. We set the number of distractor objects inside the sphere to be 16, 64 and 256, yielding a total of 3, 4, and 5 clicks to select the target with SQUAD in each distractor density condition. This leads to a theorized $3c+v_{t_0}$, $4c+v_{t_1}$ and $5c+v_{t_2}$ seconds to select a target, where v_{t_i} is the total visual search time across all refinement phases in each condition. The value of c needs to be empirically determined, but we expect it to be less than one. Again, we believe v_{t_i} to be low and not affect movement time by a large amount.

4.3 Empirical Evaluation

In order to empirically validate the results from our analytic evaluation, we performed a comparative study of SQUAD and standard ray-casting.

4.3.1 Apparatus

We used a back-projected VisBox-SX system, with only one projector (monoscopic) to display the experimental environment on a $2.29m$ x $3.05m$ screen. The resolution of the graphics was $1400px$ x $1050px$. A wireless Intersense IS-900 Wand was used for controlling the cursor on the screen.

The experimental software was written using the Vizard Virtual Reality Toolkit by WorldViz. It ran under Microsoft Windows XP on a workstation with an Intel Core2 660 CPU at 2.40GHz and 2GB of RAM. The frame rate was fixed at 55 frames per second for all conditions except with the high-density distractor conditions, in which it went down to around 15 frames per second in the sphere-casting phase only, because many collision tests with the selection sphere were necessary. We were comfortable with the drop in frame rate for that one condition because the sphere-casting selection was very easy to perform.

The environment consisted of circular objects as shown in Figure 4. The user stood at the center of an invisible sphere of $2.155m$ radius, at an orthogonal distance of $1.52m$ to the display surface. The red target and the gray distractors were evenly distributed on the surface of the sphere. There was no head tracking or any virtual navigation of the environment and the user remained at a fixed location. The perspective projection of the objects caused them to have the correct visual size from the user's point of view at the center of the sphere. We made the decision to render the circles on the surface of a virtual sphere, as opposed to on the flat screen plane, because the effective angular width of objects displayed far from the center of a flat screen decreases [9]. The perspective rendering of the circles near the edges of the display compensated for this effect in our environment, such that all objects had the same angular width from the user's point of view.

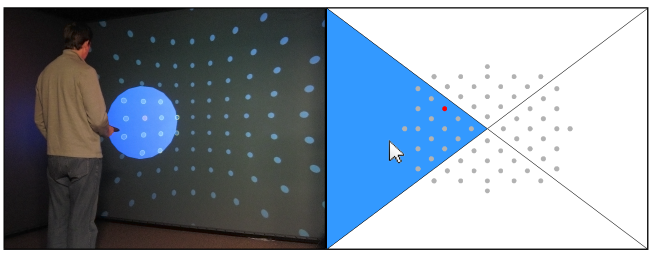

Figure 4: Left: Experimental setup with sphere-casting. Right: quad-menu stage of SQUAD in the experiment.

The target position was randomly selected from a list of candidate targets that fell inside a torus-shaped section of the display sphere, and was limited by a small radius of $0.52m$ and a large radius of $0.77m$. This ensured that targets were presented in all directions from the center of the projection.

The cursor position was determined by a function of the yaw and pitch of the IS-900 wand and the display's field of view. With the user standing at a fixed position in front of the display, the position of the cursor closely matched the ray extending from the wand. We decided to rely only on the angular readings of the wand, rather then implementing 3D ray-casting based on the combination of position and orientation information, because we wanted to keep the motor difficulty to complete the task constant. Participants were told to keep their hand position within a small range over a mark on the floor that determined the center of the virtual sphere, and not to reach out with their arms. With the cursor position dependent solely on the wand's yaw and pitch, we were able to keep the motor behavior identical to that of ray-casting from the sweet spot at the center of the virtual sphere. There was then, of course, a mismatch between the position of the displayed cursor and that of the position of the 3D ray extending from the user hand with the screen. This offset was, however, minimal and no participant seemed to mind, or even notice, the difference.

Each task began with only two objects on the screen: a large yellow circle in the center, and the red target. Once the user clicked the large yellow object, it disappeared and the rest of the screen was filled with distractor objects. We did this for two reasons. First, clicking at a pre-determined spot before the start of a task meant that the angular amplitude of the movement was kept in a controlled range. Second, by not showing the distractors in the beginning, the user could find the target location before starting the task, reducing any cognitive time to segment the target from the distractors to a minimum.

For the ray-casting condition, a crosshair represented the cursor and the task was finished when the user clicked the trigger button on the IS-900 wand. When the cursor intersected with an object, either the target or a distractor, the object was highlighted with a yellow border.

In the SQUAD condition, after the user clicked at the yellow object in the center of the display to begin the task, the cursor changed to a sphere (Figure 4, left). All objects that were inside or intersecting with the surface of the sphere were rendered with a highlight, indicating that they were active for selection. The sphere-casting action was committed by a click with the trigger button, and the display changed to the quad-menu (Figure 4, right). In order to maintain experimental control, for each distractor density, the quad-menu contained the same number of elements, even if the sphere did not have exactly that number of objects inside. These numbers were close enough that no participant ever noticed a mismatch between the objects inside the sphere and the objects displayed in the quad-menu. In the quad-menu, we decided to limit the display of the objects to approximately 50° of the view-field, as opposed to the full 90° of the projection screen. We made this decision to minimize the potential visual search time after the menu was displayed, and we found that 50° was enough to display a large number of objects, while still allowing the user to spot the target without any head movement. To refine the quad-menu selection, the user only needed to point anywhere in the quadrant that contained the target and click the trigger button.

We applied a dynamic recursive low-pass filter [21] to the raw pitch and yaw data from the IS-900 wand. This filter provided a rapid response time while reducing tracking jitter to a minimum (the Kalman filters provided by the IS-900 system had a significant lag when the cursor was moving precisely, causing strange cursor "stickiness" effects).

For the jittery conditions, we applied a random offset between -0.21° and 0.21° at each frame to the filtered yaw and pitch readings of the wand. This resulted in a maximum error of 80% of the smallest target width, such that participants had a reasonable chance of successful selection in the hardest ray-casting condition.

4.3.2 Participants

We recruited 16 voluntary unpaid participants from the campus community to perform the study. Participants' ages ranged from 20 to 31 years old, with a median age of 22.5. Nine of the participants were female.

4.3.3 Procedure

Upon arrival, participants were greeted by the experimenter and given an informed consent form to read and sign. They were then given a color blindness screening test and proceeded to complete a background questionnaire. After that, they were shown the experimental setting and started learning the first technique *vs.* tracking combination. The learning was done with an easy condition so they could understand the technique without making errors. They were then given a practice session, in which they had to practice all nine target size *vs.* distractor density combinations in the current condition for at least $90s$.

After practicing, they were reminded that they had to perform the trials as quickly as possible while trying not to make errors, and then performed eight sets of each of the nine combinations. When errors were made, the application displayed a message ("Not quite!") for $0.7s$ and the next task was displayed. The erroneous trial was then put into an array of trials that was presented in a new random order after the end of the set of trials for the current technique *vs.* tracking combination. This process was repeated to a maximum of five attempts per trial. If the user made five errors on a trial, it was deemed failed and was not presented again. The target position was the same for all attempts of a given trial.

After the end of each technique *vs.* tracking session, the participant completed a set of rating-scale questions and rested for up to two minutes. They then moved on to the next condition, following the same protocol, until all four technique *vs.* tracking combinations were completed.

Finally, the participant filled out a post-hoc questionnaire, comparing both techniques overall and in light of the other variables.

4.3.4 Results

We performed a factorial ANOVA with repeated measures on both dependent variables: time to complete a task and mean number of errors per trial.

Time Overall, ray-casting was significantly faster than SQUAD ($F_{1,15} = 4.92$, $p < 0.05$), but only by two-tenths of a second. Interestingly, there was no main effect of tracking ($F_{1,15} = 0.001$, $p = 0.979$).

There were main effects of both distractor density ($F_{1,15} = 398.6$, $p < 0.0001$) and target size ($F_{1,15} = 153.4$, $p < 0.0001$). This indicates that the effects of distractor density on SQUAD and target size on ray-casting were so large that the variables were significant overall, but when we examine the interactions, we get a clearer picture of the effects.

The tradeoff of a single precise selection compared to multiple coarse selections can be clearly seen in the interactions of technique with target size and distractor density. Figure 5 shows the significant interaction between technique and distractor density ($F_{2,30} = 290.51$, $p < 0.0001$). The 95% confidence interval, at each density level, showed that SQUAD was significantly faster in the low density, that there was no significant difference for the medium density, and that ray-casting was significantly faster with high density.

As expected, all densities were significantly different from each other with SQUAD, while there was no statistical evidence for a difference for ray-casting between any distractor density pairs. There was, however, a slight increase in the mean task completion time for ray-casting as the distractor density increased. We believe that this may have been caused by an increase in visual processing time, as the distractors were highlighted as the cursor intersected with them. However, further studies should be done to verify this effect.

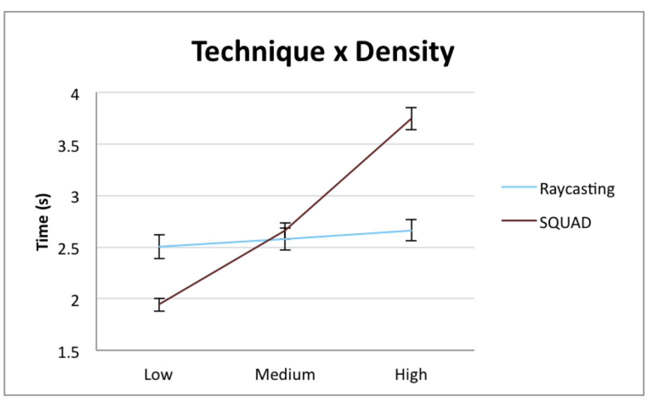

Figure 5: Interaction between technique and distractor density. The error bars represent standard error.

The other interaction that evidences the tradeoff is that of technique with target size. There was a highly significant interaction of these factors ($F_{2,30} = 135.17$, $p < 0.0001$), illustrated by Figure 6. As expected and predicted by the distal pointing model, pairwise comparisons showed that the smallest targets took significantly longer to select with ray-casting, while there were no significant differences among target sizes for SQUAD.

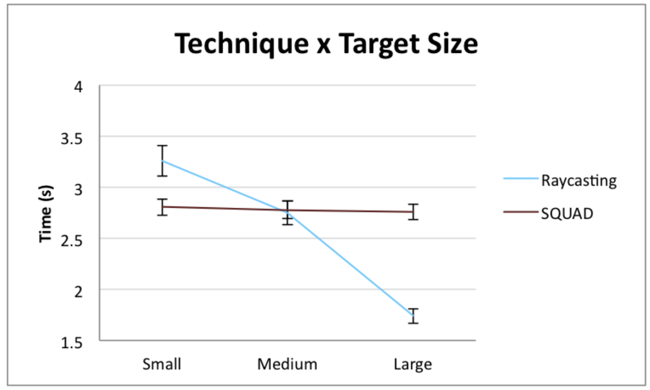

Figure 6: Interaction between technique and target size. The error bars represent standard error.

Looking at the interaction between technique and tracking, we expected to see that ray-casting would be slower with jittery tracking, while SQUAD would not be affected by tracking jitter. However, this interaction was not significant at a 95% confidence level ($F_{1,15} = 3.93$, $p = 0.066$). Despite near-significance, the mean difference in time with ray-casting was only about 0.1s, and more errors were made with bad tracking, which could indicate that participants favored speed over accuracy, even if instructed otherwise.

No other significant interactions were found, which is consistent with our hypotheses.

Figure 7 shows the mean results for all technique-density-target size combinations (the three densities for ray-casting are averaged in this graph, since density had no effect on ray-casting performance). It is clear from this graph that SQUAD was significantly faster than ray-casting with low density and either small or medium size targets, and with medium density and small targets. In two other conditions, there was no significant difference between the two techniques. Finally, there are four conditions where ray-casting is significantly faster than SQUAD.

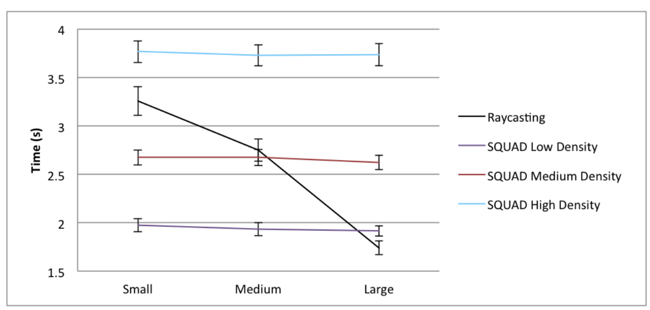

Figure 7: Mean results for all technique-density-target size combinations. Note that all ray-casting densities are averaged and displayed in a single line, since there was no significant difference among them. The error bars represent standard error.

Errors As expected, there was a significant main effect of technique with respect to errors ($F_{1,15} = 56.86$, $p < 0.0001$), with more errors being made with ray-casting. In fact, virtually no errors were made with SQUAD. The overall error rate with this technique was 0.007 errors per trial.

The lack of errors with SQUAD makes it interesting to look at the effects of tracking, target size and distractor density on ray-casting. Thus, we performed a new repeated measures ANOVA removing all the SQUAD conditions.

As expected, there was a significant main effect of target size on the number of errors per trial with ray-casting ($F_{2,30} = 46.21$, $p < 0.0001$), with more errors made with the smallest targets.

Although the average number of errors was higher for the jittery conditions, we found no statistical evidence of this difference ($F_{1,15} = 1.12$, $p < 0.31$). We believe that, since the amount of jitter was controlled, users were able to learn and compensate for it, since the low pass filter applied before the jitter allowed most participants to keep the cursor fixed on an exact pixel when needed. That, combined with the fact that the maximum jitter was 80% of the minimum target width and the continuous clear highlighting feedback of cursor intersection may have caused users to adapt and learn to select accurately with ray-casting despite the jittery cursor.

User preference SQUAD was largely preferred by all participants of the experiment. When asked which technique they favored overall, when the cursor was jittery and when the targets were small, all 16 participants answered SQUAD. When asked about which technique they preferred when there were many distractors in the scene, the majority (nine) still preferred SQUAD, suggesting that the increased number of steps did not outweigh the overall preference of the technique; two participants were undecided, and the remaining five preferred ray-casting when many distractors were present.

It is also interesting to look at subjective ratings the participants gave for various aspects of both techniques. Participants were instructed to fill out a survey immediately after completing each of the techniques, and to rate the techniques on a seven-point scale for ease of learning, ease of use, and how hard the techniques were in various conditions (when the cursor contained artificial jitter, the targets were small, and there were many distractors). Also on a seven-point scale, participants were asked to rate their wrist, leg and back fatigue. Before answering the survey after the last technique, participants were instructed to respond independently of the answers to the first one.

We performed Wilcoxon Signed Rank tests on each of the questions. There was no significant difference in the reported ease of learning between the techniques ($n = 7, W = 20, insignificant$). For ease of use, ray-casting ($mdn = 4.5$) was ranked significantly more difficult than SQUAD ($mdn = 1$) ($z = 3.16, p < 0.001$). Participants found ray-casting significantly more difficult when the cursor was jittery ($z = 3.24, p < 0.001$) and when the targets were small ($z = 3.5, p < 0.001$). There was no significant difference with respect to task difficulty when many distractors were present ($mdn_{ray-casting} = 2, mdn_{SQUAD} = 3.5, z = -0.18, p = 0.19$).

Participants reported significantly more arm fatigue with ray-casting ($mdn = 5$) than with SQUAD ($mdn = 3.5$) ($z = 2.65, p < 0.05$). No significant difference was found between the two techniques for leg ($n = 5, W = 5, insignificant$) and back fatigue ($mdn_{ray-casting} = 3, mdn_{SQUAD} = 2, z = 1.52, p = 0.064$).

4.4 Model Validation

Based on the analytic evaluation and the empirical results of both techniques, we can validate the predictive models for the ray-casting and SQUAD pointing tasks.

Figure 8 *left* shows the close linear fit of the ID_{DP}'s of each ray-casting conditions based on the distal pointing model to the actual task performance time. However, we note that the intercept of the regression line is quite a bit higher than that predicted by the original model proposed by Kopper *et al.* [9]. We believe that this is due to the nature of how errors were considered in the two experiments. In Kopper's experiment, errors did not invalidate a trial, so participants could be more careless when trying to select a target, as they could click multiple times to achieve the selection. In our case, on the other hand, the whole trial had to be attempted again, so we believe participants were more careful and certain that the cursor was inside the target area before they clicked. This resulted in a higher minimum time to complete a trial. The slope of the regression line is quite similar (0.028 in the original model and 0.037 in our experiment), and the correlation coefficient (R^2) is as high as 97.5%, which provides evidence that the distal pointing model was valid in our experimental environment.

We can analyze the SQUAD pointing trials based on the time it took for each of the phases, which consisted of sphere-casting followed by two, three or four refinements. Overall, as we predicted, the selection time has a linear relationship with the number of refinements, as shown in Figure 8 *right*. Notice that the growth is linear and the intercept is very close to zero, which emphasizes the constant increase in time as more refinements are needed.

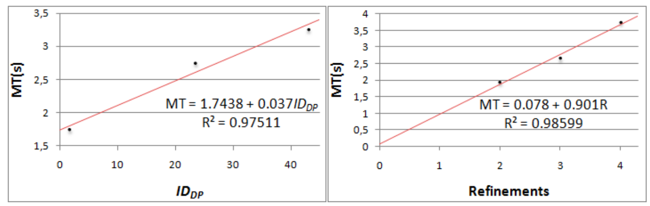

Figure 8: Scatter plot and regression line for the (*left*) ray-casting and (*right*) SQUAD pointing conditions.

Interestingly, there was a significantly longer time for the quad-menu selection in the first refinement step of the high-density distractors condition, in which there were a total of 256 objects in the quad-menu. The difference was on the order of $0.2s$ longer than in any other refinement phase, which were all within $0.05s$. The conclusion we derive from this is that visual search time was only meaningful when there were a very large number of objects in the quad-menu, while in all other conditions, this time was negligible. However, the time to complete the first refinement phase was significantly higher for all three target densities.

5 Discussion

The analytical evaluation of SQUAD selection was backed up by the results of an empirical study comparing it to standard ray-casting. We verified that the there is, indeed, a performance tradeoff between immediate techniques that use one precise action to select an object and progressive refinement techniques that require very low precision at the expense of multiple steps. The use of SQUAD, and, by extension, other progressive refinement selection techniques, should be based on a consideration of this tradeoff. We found that SQUAD is significantly faster for selection of small objects and selection in low-density environments. When errors are considered, the case for SQUAD is even stronger, as it achieved near-perfect accuracy. A positive aspect of our approach is that the increase in the number of refinements needed grows very slowly, such that the task is not likely to take many refinement phases. Another interesting aspect of SQUAD is that the time to complete a task grows linearly as the number of refinements increase, whereas with standard ray-casting, the time increase is exponential as targets become smaller.

There is potential for much further research on selection by progressive refinement techniques. SQUAD is only one of a large set of techniques that fall within the design space of progressive refinement techniques. While SQUAD was highly efficient with near-zero error rates and better time than standard ray-casting with small targets, it has some limitations that need to be acknowledged. SQUAD was designed with a particular application setting in mind, and the nature of the task, which involved distant objects roughly arranged on a surface, influenced the design of SQUAD. Its sphere casting component is not well-suited for selecting from among items distributed in depth. However, SQUAD can be adapted to work well in such situations. For example, instead of a selection sphere, a cone or cylinder could be used to specify a deeper region of initial selection for further refinement in the quad-menu phase.

In addition, SQUAD works well for tasks that require the selection of objects that are visually distinct from other possible objects in the vicinity, and that do not depend on the spatial context. Selection tasks that depend on object location rather than visual features, for instance, cannot be achieved by SQUAD, but still may be achieved effectively with selection by progressive refinement. For example, objects can be kept within their spatial context if refinement is accomplished by zooming. In this case, the refinement would consist in the specification of an area in the view that would zoom to fill the display, decreasing the number of selectable objects. This is not necessarily equivalent to navigating close to objects to select them; the zooming could be done discretely. After the selection task, the viewpoint could return to the original position.

We found that the distal pointing model proposed by Kopper et. al [9] accurately predicted performance with ray-casting. We also were able to find evidence that the performance of discrete progressive refinement selection techniques can be modeled by a direct linear relationship to the number of refinements necessary for completing a task. The use of analytical models in the evaluation of such techniques can provide a benefit in more realistic settings, in which control can be traded off for ecological validity. In such situations, many aspects, such as user strategy, confound the experimental control, and using reliable analytical models in the performance assessment may be the best choice.

6 Conclusion and Future Work

We have introduced the concept of selection by progressive refinement and proposed a design space. We designed a progressive refinement technique, Sphere-casting refined by QUAD-menu (SQUAD), and evaluated it against standard ray-casting. The results indicate that there is a tradeoff between the number of refinements and the required pointing accuracy that must be taken into account for the design of 3D selection techniques. When the visual size of objects is too small and the density of the environment is not too high, selection can be achieved more efficiently by progressive refinement. Furthermore, progressive refinement techniques can ensure very high levels of accuracy because they do not require precise pointing.

We plan to continue this research by designing and evaluating additional progressive refinement techniques. In addition, we plan to compare SQUAD and similar techniques to high-precision pointing techniques such as PRISM.

References

[1] D. A. Bowman, E. Kruijff, J. J. L. Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004.

[2] P. Figueroa, Y. Kitamura, S. Kuntz, L. Vanacken, S. Maesen, T. D. Weyer, S. Notelaers, J. R. Octavia, A. Beznosyk, K. Coninx, F. Bacim, R. Kopper, A. Leal, T. Ni, and D. A. Bowman. 3DUI 2010 contest grand prize winners. *IEEE Computer Graphics and Applications*, 30:86–96, 2010.

[3] C. Forlines, R. Balakrishnan, P. Beardsley, J. van Baar, and R. Raskar. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 73–82, New York, NY, USA, 2005. ACM Press.

[4] S. Frees, G. D. Kessler, and E. Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Trans. Comput.-Hum. Interact.*, 14(1):2, 2007.

[5] T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–290, New York, NY, USA, 2005. ACM Press.

[6] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2006. ACM Press.

[7] G. D. Haan, M. Koutek, and F. H. Post. Intenselect: Using dynamic object rating for assisting 3D object selection. In *Virtual Environments 2005*, pages 201–209, 2005.

[8] M. W. C. Jr and S. C. Fullenkamp. Perceived contrast and stimulus size: Experiment and simulation. *Vision Research*, 28(6):695 – 709, 1988.

[9] R. Kopper, D. A. Bowman, M. G. Silva, and R. P. McMahan. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies*, 68(10):603 – 615, 2010.

[10] J. Liang and M. Green. Jdcad: A highly interactive 3D modeling system. In *3rd International Conference on CAD and Computer Graphics*, pages 217–222, August 1993.

[11] J. Lucas. Design and evaluation of 3D multiple object selection techniques. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2005.

[12] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.

[13] M. R. Mine. Virtual environment interaction techniques. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[14] M. R. Mine, J. Frederick P. Brooks, and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[15] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3D immersive environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff., New York, NY, USA, 1997. ACM.

[16] A. Steed. Towards a general model for selection in virtual environments. In *Proceedings of the 2006 Symposium on 3D User Interfaces.*, pages 103–110. IEEE, 2006.

[17] A. Steed and C. Parker. 3D selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In *8th International Immersive Projection Technology Workshop*, pages 13–14, 2004.

[18] A. Steed and C. Parker. Evaluating effectiveness of interaction techniques across immersive virtual environmental systems. *Presence: Teleoper. Virtual Environ.*, 14:511–527, October 2005.

[19] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[20] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. *3D User Interfaces*, 2007.

[21] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 33–42, New York, NY, USA, 2005. ACM Press.

[22] S. Zhai, J. Kong, and X. Ren. Speed–accuracy tradeoff in Fitts' law tasks–on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 61(6):823 – 856, 2004.

[23] S. Zhao, M. Agrawala, and K. Hinckley Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1077–1086, New York, NY, USA, 2005. ACM Press.