DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments

Annette Mossel[1] Benjamin Venditti[1] Hannes Kaufmann[1]

[1]Interactive Media Systems Group, Vienna University of Technology

Favoritenstr. 9-11/188/2, 1040 Vienna, Austria

mossel@ims.tuwien.ac.at

ABSTRACT

One of the primary tasks in a dense mobile augmented reality (AR) environment is to ensure precise selection of an object, even if it is occluded or highly similar to surrounding virtual scene objects. Existing interaction techniques for mobile AR usually use the multi-touch capabilities of the device for object selection. However, single touch input is imprecise, but existing two handed selection techniques to increase selection accuracy do not apply for one-handed handheld AR environments. To address the requirements of accurate selection in a one-handed dense handheld AR environment, we present the novel selection technique *DrillSample*. It requires only single touch input for selection and preserves the full original spatial context of the selected objects. This allows disambiguating and selection of strongly occluded objects or of objects with high similarity in visual appearance. In a comprehensive user study, we compare two existing selection techniques with *DrillSample* to explore performance, usability and accuracy. The results of the study indicate that *DrillSampe* achieves significant performance increases in terms of speed and accuracy. Since existing selection techniques are designed for virtual environments (VEs), we furthermore provide a first approach towards a foundation for exploring 3D selection techniques in dense handheld AR.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces| Interaction styles, I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism | Virtual Reality

General Terms

Design, Algorithms, Performance

Keywords

3D Interaction Techniques, 3D Selection, Handheld Augmented Reality, Dense Virtual Environments

Laval Virtual VRIC '13, March 20-22, 2013, Laval, France.

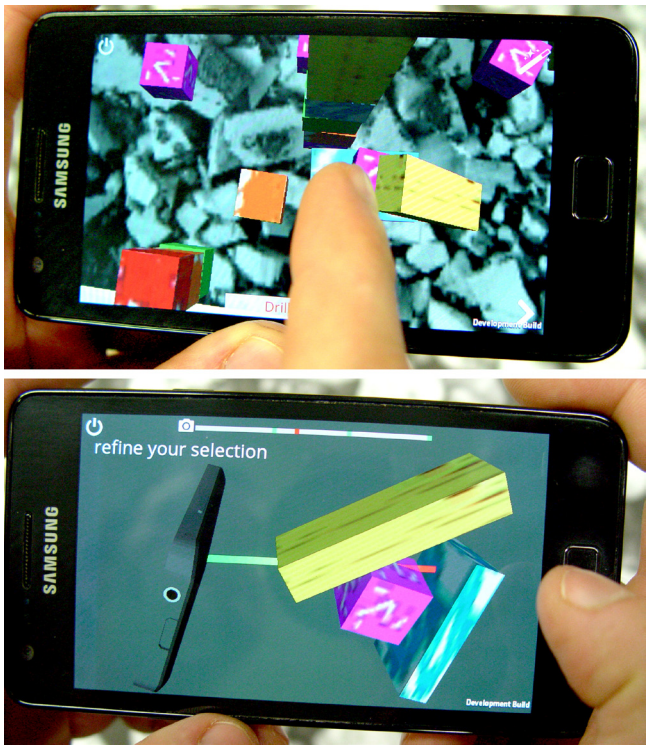

Figure 0-1: Selecting a partly occluded object (pink cube) using DrillSample selection technique

1. MOTIVATION & CONTRIBUTION

State of the art mobile devices provide real-time 3D rendering of dense virtual scenes, which are blended with the live image of the built-in mobile camera. The mobile device acts as a "window into the virtual world". In hand-held video see-through Augmented Reality (AR), 3D interaction with a virtual scene is a key feature to explore the full potential of mobile AR. One of the primary interaction tasks is 3D object selection. However, current interaction in handheld AR environments is often limited to pure 2D pointing and clicking via the device's touch screen. Using (multi) touch input for 3D object selection induces several problems. Since only one hand is available for interactions in a typical one-handed handheld AR setting, only simple touch gestures of one hand for selection are suitable. Furthermore, object selection using touch input can be imprecise due to the large area the user's fingertip covers on the screen. This is especially an issue when selecting an object in dense virtual environments where a large number of tightly grouped virtual objects form the scene. Therefore, the selection technique must provide the possibility to disambiguate and precisely select desired objects even if they are partly occluded or completely

hidden by other virtual scene objects. In dense virtual environments with objects of high visual similarity, selecting a target object can be difficult or even impossible with existing selection techniques. To provide unique selection of a target object, the technique should preserve any spatial context to support disambiguation and consist of only single touch interactions to simplify user interaction and minimize selection errors evoked by imprecise touch input.

To address these interaction problems, we present a novel selection technique *DrillSample* for dense virtual scenes. *DrillSample* is a two-step selection technique providing precise selection and disambiguation of visible, partly occluded or invisible objects, which can also be highly similar in appearance. It consists of an initial target indication step. If more than one object has been selected, it is followed by an optional refinement step where all selected objects are presented to the user as 3D virtual clones for selection refinement. There original spatial context of the selected objects in all three dimensions is preserved in this detailed visualization view. Hence, our method is fully context preserving compared to two-step selection techniques such as [1] [2]. Although *DrillSample* is introduced for precise selection in dense handheld AR environments, it can be applied to all kinds of dense VEs.

In addition to *DrillSample* we implemented two baseline selection techniques for handheld AR scenarios and integrated them into our VR/AR Framework ARTIFICe [3]. Based on this mobile evaluation framework, we conducted a comprehensive user study to evaluate accuracy, performance and usability of the three techniques. Thereby, we introduce a first approach towards a foundation for exploring 3D selection techniques in dense handheld AR.

2. RELATED WORK

Selection is one of the universal interaction tasks in a 2D as well as 3D user interface and has been extensively studied [4]. As shown in literature, performance and usability of a selection technique varies greatly, depending on specific task requirements (i.e. object size and distance) and the environment's layout such as scene density and object occlusions. It always consists of two subtasks: indicating the object and the optional step of confirming the selection. Since many selection approaches exist that have been designed for VEs, we separate related work into selection techniques designed for VEs that can be adapted to handheld AR, and selection techniques for mobile touch interfaces.

2.1 3D Selection in (immersive) VEs

To indicate the desired object, the user can occlude the target, touch it or point at it. VirtualHand [4] and GoGo [5] are selection techniques to touch objects in space. Virtual pointing metaphors include, amongst others, Ray-Casting, Occlusion, Cone-Casting, Aperture and Flashlight [4] [6] [7] [8]. Hybrid techniques exist, such as HOMER [9], which uses Ray-Casting for object selection and VirtualHand for manipulation.

Pointing techniques are generally considered to be more precise than virtual hand-based techniques, while virtual hand techniques generally perform more effectively for object manipulation tasks [4]. In dense environments, Ray-Casting is reported to perform precisely [4]. Go-Go, Aperture as well as Occlusion requires independent tracking of user's output device and another marker in space. Aperture effectively selects small objects but performs less precisely in a dense group of objects [5] [4] [7]. Occlusion is reported to be imprecise with a dense group of small or distant objects [4]. Go–Go also has low performance when selecting objects in dense environments, but can easily select fully occluded objects in a one-step selection process [4]. Fully occluded objects, however, cannot be selected in a single step with Ray-Casting, Occlusion, Aperture or Cone-Casting. Ray-Casting can select partly occluded objects due to its high precision. SQUAD [2] and Expand [1] are two-step selection techniques to aid in target selection. Internally they use Sphere- respectively Cone-Casting for target indication. In a refinement selection step, all selected objects are presented in a 2D representation preserving some spatial context information. Both techniques offer precise selection of partly or fully occluded objects in dense or cluttered environments. All those 3D selection techniques have been originally designed for input devices and usage environments other than mobile AR. Thus, their performance with multi-touch displays has not been evaluated.

2.2 Selection with Mobile Touch Interfaces

Existing approaches for selecting 3D objects in mobile AR usually use a simple pointing metaphor, triggered by a single touch event on the mobile screen [10] [11] [12]. However, in a cluttered mobile environment, these approaches lack precision due to users' finger size. To enhance precision in 2D touch interfaces, [13] 2D selection techniques that overcome the problem of finger occlusion on the screen have been presented: Dual-Finger Offset and Dual-Finger Midpoint. These approaches require two hands for precise selection and are therefore not applicable in a mobile AR scenario, where only one hand is available for input. [14] Proposes two selection techniques especially designed for handheld AR using two fingers to select small and partly occluded objects in sparse as well as dense AR. This approach is promising, but cannot be applied to select objects in the very screen corners. Since this reduces the little available interaction space even more, it is not an optimal approach on a mobile device such as a smartphone. Additionally, in [14] the authors demonstrate selection in dense mobile AR using single touch by decoupling the selection point from the physical touch point. Therefore, target object occlusion by the finger can be avoided but the offset is calculated statically. So this approach is not applicable to select objects near the screen corners and along the display edges.

Our work provides easy to use, intuitive and precise 3D selection in dense virtual environments with high visual similarity of the scene objects. *DrillSample* is valid for single device tracking and only needs one-finger input to select an object in a two-step interaction process. This design allows for precise 3D selection in one-handed handheld AR.

3. SELECTION DESIGN

When designing a selection technique for handheld AR, there are important factors that influence performance and ease-of-use.

3.1 Requirements

Since we are aiming at precise selection in dense one-handed handheld AR environments, the application scenario's specific characteristics must be taken into account during selection design as well as for baseline choice to guarantee a fair evaluation. The requirements can be summarized as follows.

(1) **Single I/O device:** Input and output device comprise a single device. Thus, independent tracking of user's interaction device and head is not available compared to (immersive) VEs. When tracking the mobile device, its six-degree-of-freedom (6DOF) pose[1] is estimated using a variety of

[1] The 6DOF pose comprises 3D position and 3D orientation

tracking methods including inside-out optical tracking (i.e. of visual features) or/in combination with the built-in mobile inertial measurement unit.

(2) **Limited gesture complexity:** Touch input by fingers can be imprecise due to the large area the user's fingertip covers on the screen. Since there is only one hand available for interaction, complex multi hand- and finger gestures cannot be applied to improve selection precision.

3.2 Design Guidelines

Based on our motivation and the outlined requirements, we developed the following design guidelines to enable precise selection in a one-handed dense handheld AR environment.

(1) **Keep direct touch abilities:** One of the most appealing aspects of touch displays is the ability to directly "touch" an object in order to select it. We aim to support this direct manner and do not introduce an offset to the cursor due to the disadvantages that are mentioned in Section 2.2.

(2) **Keep interaction simple**: Since multi finger interaction is not a straight forward metaphor and requires prior knowledge of specific gestures, we aim to reduce user touch input complexity for object selection. Only one-finger input should be applied to allow precise object selection. Two-finger input using a single hand should only be applied for optional interaction such as detailed inspection of selected objects.

(3) **Enable disambiguation and unique selection:** Since objects can be partly occluded or even invisible in dense virtual scenes, it is important to provide a technique that supports selection of these objects. Furthermore, objects can be highly similar in visual appearance. Thus, it is important to present multiple selected objects in a correct spatial context that assists object disambiguation while taking the limited screen size into account.

3.3 Baseline Techniques

Most of the popular 3D selection techniques mentioned in Section 2.1 are not designed for mobile environments. Many of them, such as Go-Go or Aperture require independent tracking of the user's interaction device and head. Furthermore, popular multi-touch selection techniques aim at selecting objects in a 2D environment. Hence, direct comparison of these techniques is difficult in a handheld environment. Related work [14] introduces a qualitative evaluation of 3D selection techniques in handheld AR environments. For performance analysis, they propose an adaption of Go-Go using swipe gestures to adjust the virtual arm length and multi-touch input to select an object. However, this adaption changes the direct mapping between hand, arm and target of the original algorithm and does not apply for a clean and fair performance evaluation of selection techniques in handheld AR. To compare our novel selection technique with other baseline selection techniques for a summative evaluation, we chose *Ray-Casting* [4] and *Expand* [1] because they represent techniques with different numbers of selections steps, are both suitable for dense environments and can be adapted to handheld AR without changing the original mapping characteristics during interaction. Both fulfill the requirements from Section 3.1.

Ray-Casting is a simple, one step selection technique and is widely used in 3D computer and (immersive) VEs. A virtual ray is cast into the virtual scene; objects are selected if the virtual ray intersects them. *Ray-Casting* is fast and accurate for objects in close range, but has problems with small objects in greater distance and with occlusions. To use Ray-Casting in a handheld AR environment, we use the following adaption: *Ray-Casting* is triggered by a single touch event one the screen; these 2D screen coordinates are back-projected into 3D space and a virtual ray is cast from the virtual camera's position in direction of the back-projected 3D point into the handheld AR scene. This direction can be estimated using the handheld's device 6DOF pose that is implicitly given in handheld AR. The first object the ray hits is selected. Hence, object selection results in a simple single touch experience, which should be easily understandable for users.

Expand is a two-step technique in which virtual scene objects are selected using Cone-Casting [6]. In a second step, objects cast by the cone are presented in a virtual grid for accurate selection. This technique was designed to work in conditions where there are many objects within the cursor position. *Expand* features spatial context preservation in the x/y-domain in the refinement step as well as the possibility to accurately select partly or completely occluded objects.

3.4 DrillSample Selection Technique

Following the design guidelines from Section 3.2, we propose the two-step selection technique *DrillSample.* It requires single device tracking and only one-finger input to select an object in a two-step interaction process. Thereby, the technique allows users to uniquely identify and select visible, partly occluded or invisible objects even if they are of high visual similarity without hindering simple targeting tasks. *DrillSample* was designed in an iterative fashion derived from the two baseline approaches, which we felt would improve upon the original techniques. A formal description of the *DrillSample* selection technique is given in Figure 3-1.

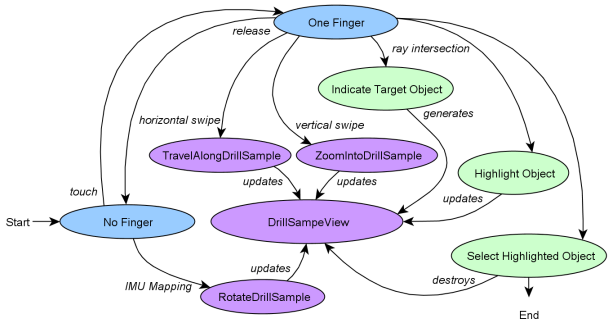

Figure 3-1: State diagram for *DrillSample* selection

In the first step, the user indicates the target in the scene using single touch. From this 2D touch point, *Ray-Casting* is performed as described in Section 3.3. Instead of selecting only the first object that the ray casts, all cast objects of the scene are selected. In the second step, all selected objects are presented to the user as 3D virtual clones for selection refinement on a solid grey background (see Figure 0-1) with the AR scene view turned off. Therefore, the ray with all clones "attached" to it is pulled out of the virtual scene, similar to a drill sample that is pulled out of the ground. Then the ray is aligned parallel to the horizontal axis of the image plane and the clones are arranged on a horizontal line, the *DrillSample* visualization. The x- and y-position of the clone correspond to the hit point of the ray with the original objects, while the depth information is represented by the clone's position on the horizontal line. The spatial context of the objects involved in the selection from the original scene layout is preserved and extends *Expand*'s approach in the depth domain. Hence we assist the disambiguation of selected objects that are occluded or of similar visual appearance.

The proposed algorithm for the *DrillSample* visualization enables inspection of the selected objects in the following ways:

1) By using the handheld's built-in Inertial Measurement Unit[2], the user can rotate around the current object of interest to inspect objects from different angles.
2) Using a one-finger horizontal swipe-like motion, the user can browse through all selected objects. The current object of interest is in the middle of the screen.
3) Using a one-finger vertical swipe-like motion (or a two-finger pinch like gesture) the user can change the distance between the *DrillSample* to the virtual camera to either zoom-in to a particular object or zoom-out to get an overview. This interaction is especially helpful on small displays to gain a quick overview if many objects have been selected.

Finally, the user selects the desired object using one-finger touch input. Upon selection from the *DrillSample* visualization, the clones are destroyed; the user is informed of his selection and the application switches back to handheld AR scene view. To avoid confusion with the original AR scene and to further underline the refinement with the decoupled *DrillSample*, the AR scene view is turned off during refinement (see Figure 0-1). The workflow is illustrated in Figure 3-2.

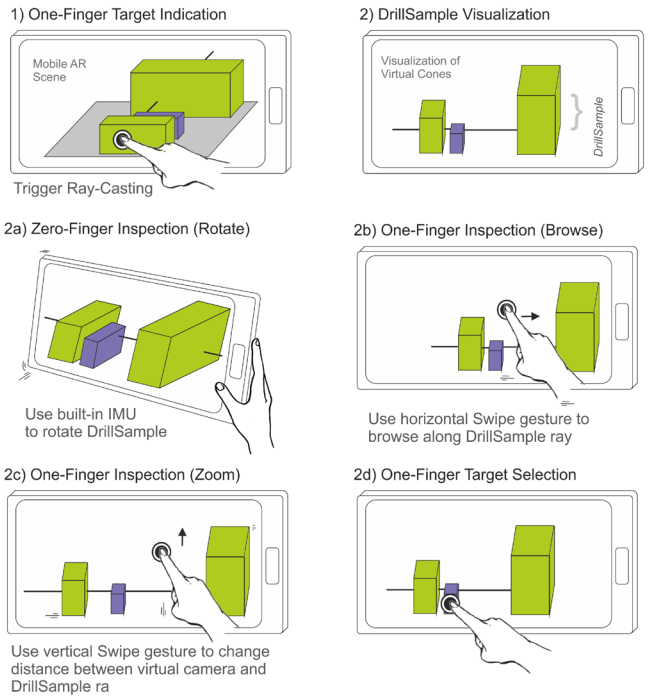

Figure 3-2: *DrillSample's* two-step selection process

DrillSample is especially useful in dense environments but also works well in sparse scenes when only single objects are selected. While the selection process takes extra time if the two-step process is performed, single object selection has been optimized. If a single object is detected along the ray, the object is automatically selected and the second step is bypassed.

[2] The IMU consists of accelerometer, gyroscope and possibly magnetometer.

3.4.1 DrillSample Algorithm

To formalize the illustrated selection process, our proposed *DrillSample* algorithm can be denoted as follows:

Step 1: Target indication & DrillSample construction

(1) The user performs a touch on the device's screen at $p_T \in \mathbb{R}^2$.
(2) Perform Ray-Casting along $\vec{v}(P_{VC}, dir) \in \mathbb{R}^3$ into the virtual scene, with $P_{VC} \in \mathbb{R}^3$ being the virtual camera's position and dir the direction from $P_{VC} \in \mathbb{R}^3$ to the back-projected point $P_T \in \mathbb{R}^3$ of the touch point $p_T \in \mathbb{R}^2$ in screen space.
(3) Create clones for each object that got hit, storing its original orientation, visual appearance and hit-point.
(4) Optimize the length of the *DrillSample* (see 3.4.2.1).
(5) Calculate a pivot point in at the center of all hit-points.
(6) Rotate the clones around the pivot point's vertical and z-axis, so that the *DrillSample* lies parallel to the image plane's horizontal axis. Objects hit first are then on the left and those hit last on the right side of the *DrillSample*.
(7) Z-Positioning of the *DrillSample* (see 3.4.2.2).

Step 2: Optional inspections during refinement step

(1) Rotate the *DrillSample* by mapping the device's gyroscopic sensor values to a pivot point.
(2) Calculate touch points and direction of the horizontal swipe gestures to travel along the *DrillSample* if it spans multiple screens.
(3) Calculate touch points and direction of the vertical swipe gesture (or optional two-finger pinch gestures) to change the distance between the virtual camera and the *DrillSample* (see 3.4.2.2).

Step3: Final target selection

(1) Select object by using single touch point's coordinates and *Ray-Casting* as described in step 1-1 and 1-2.
(2) Destroy clones and *DrillSample* ray.

First tests revealed to better restrict some rotations of the device applied to the *DrillSample* in step 2-1, since they were not beneficial for the users' perception of the spatial context or even confusing. Most important, all rotations around the roll axis and rotations around the pitch axis in [180°, -180°] should be discarded. Thereby, the *DrillSample* is always aligned to the horizontal screen with the first object hit positioned on the left side (see Figure 3-2). Furthermore, the 1:1 mapping between device and *DrillSample* orientation proved to be too cumbersome to inspect the objects from their back sides. Therefore, it was found to be useful to speed up the rotation around the yaw axis 3-times and around the pitch axis 1.5-times.

3.4.2 Critical Aspects

To provide a reasonably refined visualization of the virtual clones, there are two critical aspects:

(1) The length of the *DrillSample* ray needs to be optimized while preventing intersection of the clones and preserving their relative distances.
(2) The optimal Z-Position of the *DrillSample* to the virtual camera must be obtained.

3.4.2.1 Length of the DrillSample Ray

Since the relative distance of objects to each other is sufficient to preserve the spatial context, the real length of the ray should be scaled for its visualization to provide an optimal overview. If the objects are far away from each other, the ray might be shortened, or stretched to reveal objects that are inside of another (e.g. a ball

in a bucket). The optimal amount by that the ray should be scaled depends on the shortest distance between the convex hulls of the two neighboring objects along the direction of the ray. For objects with overlapping hulls, the distance is specified as a negative value and positive otherwise. Assuming n objects on the *DrillSample* and the shortest distance between $(n-1)$ neighbors is denoted by d_i, the length of the ray is then modified by $-d_i * (n-1)$.

To precisely calculate these distances can be computational costly, especially in dense environments with complex shapes. To minimize the computational load, we chose an approximation with linear complexity by treating all objects as spheres (see Figure 3-3) with the maximum extent of the objects' bounding box used as its radius and the hit point as its center.

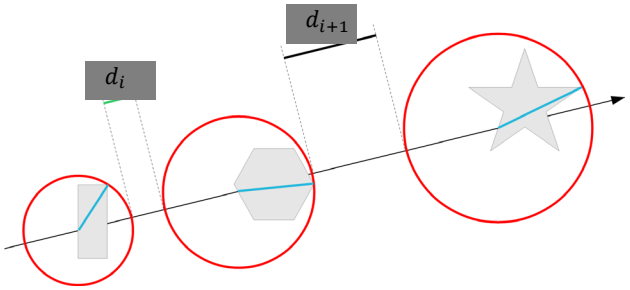

Figure 3-3: Sphere approximation of clones' size to calculate the optimal ray length

For objects whose center point is not close to the ray or have a complex concave shape, this may not be visually pleasing as it overestimates the real distance between neighboring objects. More elaborate algorithms can be employed in the future to enable an optimal adjustment of the length.

3.4.2.2 Z-Position of the DrillSample

In the current algorithm we present an overview of all selected objects. This results in the following problems.

(1) **The larger the distance between clones varies**, the less the *DrillSample* ray can be compressed. To provide an overview of all clones on one screen, the ray must be positioned at greater distance to the virtual camera. This could result in small clones being barely visible.

(2) **The more the size of the clones varies**, the less likely there is a distance to the virtual camera at which all clones are nicely visible. Small objects may appear too small or big objects might be clipped at the near image plane.

(3) **The more objects are selected**, the less likely the overview provides a meaningful starting point for refinement, as the clones in the overview appear too small as in (1).

Thus the distance between the virtual camera and the *DrillSample* depends on the size of the clones and their relative distance to each other. The overview distance D_{ov} of the virtual camera to obtain an overview can be calculated by:

$$D_{ov}(B_{DSS}) = \frac{exp}{tan(fov * 0.5)} + B_{DSS}(z), \qquad with$$

$$if\left(R_B < R_{fov}\right) \quad then \quad exp = B_{DSS}(y);\ fov = fov(y)$$
$$else \quad then \quad exp = B_{DSS}(x);\ fov = fov(x)$$

While $B_{DSS} \in \mathbb{R}^3$ is the *DrillSample's* axis-aligned bounding box represented as an expansion vector, R_{fov} is the aspect ratio of the virtual camera's field of view, R_B the aspect ratio of the bounding box's side facing the camera and fov is the field of view of the virtual camera. Additionally it has to be ensured, that neither the near nor the far clipping plane of the virtual camera are violated. The interval, in which users may modify the distance of the camera to the *DrillSample* with a vertical swipe gesture (see 2c in Figure 3-2), is then limited to $[D_{ov}(B_C), D_{ov}(B_{DSS})]$ by the bounding box of the biggest clone B_C on the *DrillSample*.

However, the *optimal* distance depends on the specific application.

3.4.3 Enabling Technologies & Algorithm Variations

DrillSample is originally designed for multi-touch displays which allow for one-finger or two-finger tracking. Since tracking of two independent contacts of the surface is only necessary for optional interactions in the *DrillSample* visualization view, the algorithm is not limited to one-handed handheld AR.

It can be easily applied in different scenarios such as desktop, semi- or fully-immersive VEs with high object density and visual similarity of the scene objects. In immersive VEs, the *DrillSample* visualization does not depend on display size, but on the field of view (FOV) of the user's output device, such as a Head Mounted Display (HMD). Instead of swiping through (multiple) selected objects, the user could instead walk along and around the ray for detailed inspection. Furthermore, the rotation of the user's virtual hand can be mapped to rotate the *DrillSample* ray in the detailed view.

4. PERFORMANCE STUDIES

For a comprehensive evaluation of our proposed selection technique, we conducted a summative evaluation by comparing *DrillSample* selection technique with the two baseline techniques mentioned in Section 3.3 across three different selection scenarios based on variations of object density and visibility.

4.1 Objectives

The main goal of the experiment was to evaluate the performance and ease of use of *DrillSample* compared to competing techniques. In this study, we focused on selection of objects in closer range in dense environments. A second objective is to examine the performance of the spatial context preservation of our proposed algorithm in environments with objects of high visual similarity. In designing the experiment, we formulated the following hypothesis:

[H1] *Ray-Casting* will be best suited for non-occluded objects.

[H2] *Expand* and *DrillSample* will perform considerably better than *Ray-Casting* in environments with overlapping, partly occluded or invisible objects, which differentiate significantly in appearance, in terms of speed and precision.

[H3] *Expand* will suffer in environments with objects of high visual similarity. Likewise, *DrillSample* will perform considerably better than *Expand* in terms of speed and precision.

4.2 Experimental Design and Procedure

We conducted the study using a within-subjects factorial design where the independent variables were manipulation technique and task scenario. The selection techniques were *Raycasting, Expand* and *DrillSample*, while the scenarios included three different experimental tasks with varying selection conditions in close range. The dependent variables were number of selections and overall task completion time. Furthermore, we measured user preferences for both technique in terms of speed, accuracy, and usability.

The user study consisted of a pre-questionnaire followed by a practical test and a post-questionnaire. It took approximately 25 minutes for each participant to finish the user study. At the beginning of the study, each participant was asked to read and sign a standard consent form and to complete a pre-questionnaire. We asked standard questions about age, gender and prior experience. Upon completion, the participant was given a detailed description of the practical part about "Selection in Handheld AR". A tutor coached them on how to use the handheld device and how to perform selection in the testing environment. Afterwards, each participant had five minutes time to practice the three selection techniques. Once they started the study, they were not interrupted or given any help. Upon completion of the practical part, they were asked to fill out a post-questionnaire (see Table 1).

Of the 28 participants ranging from 23 to 38 years, 12 were female and 16 male. 12 users had no experience of playing mobile 3D games and 7 had no experience with smartphones. One person reported to have occasionally severe pain issues in her/his primary hand's wrist. All 28 participants yielded successful simulation trials from which all data was used for analysis.

Table 1: Post-Questionnaire

Q1	How adequate do you feel the time allotted for practice was?
Q2	How comfortable were you with using a smartphone for task completion?
Q3	How would you rate the RAYCAST selection technique in terms of usability? Speed? Accuracy?
Q4	How would you rate the EXPAND selection technique in terms of usability? Speed? Accuracy?
Q5	How would you rate the DRILLSAMPLE selection technique in terms of usability? Speed? Accuracy?
Q6	Rank the three selection techniques in order of desired use (with 1 being the most desired).
Q7	When determining how much you like using a selection technique, how important in influence on your decision was usability? Speed? Accuracy?
Q8	Regarding the visualization during the refinement process of the DRILLSAMPLE technique, how helpful and useful was the linear arrangement for spatial visualization?

4.3 Implementation

All computations – tracking, rendering, selection and manipulation of virtual objects – are performed on a smartphone using Android OS. For developing and testing the proposed interaction techniques, we used the Virtual and Augmented Reality Framework ARTIFICe [3]. It uses Vuforia [15] for tracking and pose estimation of the mobile device and the Unity3D [16] game engine for 3D rendering and scene management. To access touch inputs on the mobile device screen to trigger selection and release of scene objects, ARTIFICe uses Unity's built-in Android interface to access the hardware layer. Pose data from Vuforia is processed by the specific interaction technique (IT) and handed to the ARTIFICe interaction framework for object manipulation. Using its interaction interface, we implemented *Raycasting, Expand* and *DrillSample.*

The practical test ran on a *Samsung Galaxy S II I9100*, featuring an Arm Cortex A9 Dual Core-Processor, a 4.27" WVGA multi-touch display and an 8 mega pixel camera. *Galaxy S II* weights 116g and has the physical dimensions of 125.3 x 66.1 x 8.49 mm. We protected the phone with a market available hard cover to minimize the problem of cancelling the simulations by mistake by pushing the buttons on the sides.

4.4 Tested Scenarios

We built three different scenarios to cover different selection situations in dense 3D space. They ranged from unique and un-occluded to non-distinguishable and fully occluded object selection tasks. Thus, we used occlusion and visual similarity as variables for task design. As the underlying building block [17] for interaction design, we applied the canonical task "selection", which refers to the task of acquiring a particular object from the entire set of objects available.

All scenarios are based on the same virtual working ground (black & white textured plane) that was printed to paper at 56x40cm and acted as a marker for the Vuforia framework. The marker was placed on a table that was positioned at the center of a room so that users had around 150cm of obstacle free space to work within. All 28 users completed all three scenarios in random order. Each scenario featured a simple description of the upcoming task. The participants could inspect the scenario, without being able to manipulate it, in order to understand the task according to its description before starting with the actual test.

4.4.1 Scenario1 – Unique Object & No Occlusion

The user was challenged to select a green cube in the middle of the working ground which was cluttered with around 80 other cubes of the same size but of different color (see Figure 4-1).

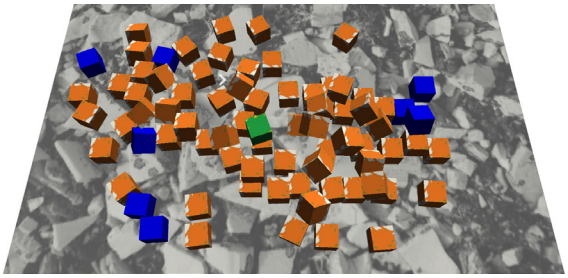

Figure 4-1: Scenario 1 "Select the *green* cube."

The targeted object was easy to distinguish and not occluded by any of the objects in the scene. As soon the user selected or confirmed the selection of the green cube, the task finished automatically.

4.4.2 Scenario 2 – Unique Object & Strong Occlusion

The user had to select a green brick in the lower right corner of a wooden textured box (see Figure 4-2). The box contained four stacks of different colored equally sized bricks.

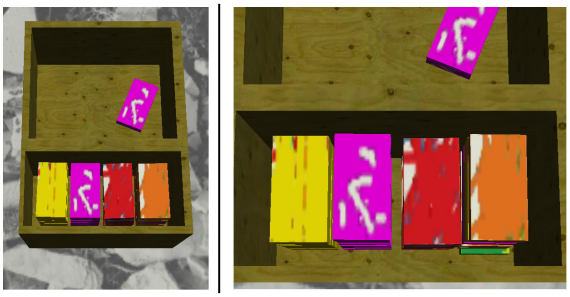

Figure 4-2: Scenario 2 "Select the *green* brick."

The targeted object was located on the very bottom of the last stack and it was the only brick that was colored in green. Although it was easy to distinguish, it was hardly visible due to the strong occlusion of the bricks stacked on top of it and the box's walls. Again, on selection of the targeted object, the task finished automatically.

4.4.3 Scenario 3 – Non-Distinguishable & Strong Occlusion

In this scenario the user hat to select a brick from a wooden textured box again (see Figure 4-3).

Figure 4-3: Scenario 3 "Select the lowermost of the pink bricks.

The box contained four stacks of equally sized bricks. All bricks were colored in light blue except for the bricks of the second stack which had a magenta colored texture. The targeted object was located on the very bottom of the magenta colored stack. It was only distinguishable by its position in the stack and was hardly visible due to strong occlusions of the bricks stacked on top of it and the box's walls. The number of bricks on top of the targeted object varied randomly for each participant from four to seven pieces.

5. RESULTS

Based on the performance study, we conducted an evaluation on the quantitative data to examine performance of the three techniques and a subjective evaluation regarding user's preferences and feedback.

5.1 Quantitative Evaluation

The quantitative data gathered from the questionnaires and automatically collected data of the test application were analyzed with Friedman's χ^2 test and repeated measures single factor ANOVA accordingly. When suitable, we employed pairwise t-tests or Wilcoxon signed rank test with the Holm's sequential Bonferroni correction. We focused on two different aspects during data analysis. First, data of all participants regarding selection techniques was evaluated and second, we analyzed the techniques' performance depending on tasks.

5.1.1 Performance Measures

Evaluating the quantitative data, "Task Completion Time" and "Number of Selection Steps" were applied as metrics. Task completion time represents the time it takes to successfully finish a specific scenario from the time, the user started it. Number of selection steps comprises the amount of necessary object selections to successfully finish a selection task. This measure indicates precision of the applied technique.

5.1.2 Performance Evaluation

The evaluation of the completion time shown in Figure 5-1 indicates significant differences for the three interaction techniques with $(F_{2,54} = 6.74, p < 0.00243)$ for all tasks on average but also with $(F_{2,54} = 9.27, p < 0.00035)$, $(F_{2,54} = 21.84, p < 1.1e-7)$ and $(F_{2,54} = 4.91, p < 0.011)$ for the tasks one to three separately.

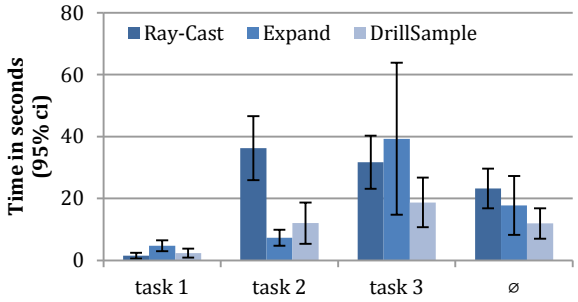

Figure 5-1: Mean completion time per task and on average

The pairwise t-test shows, that only *DrillSample* is significantly faster than *Ray-Casting* with $(t_{27} = 4.33, p < 0.00018)$ in the overall mean completion time.

For task one, the techniques *Ray-Casting* and *DrillSample* score significantly better than *Expand* with $(t_{27} = -3.82, p < 0.0007)$ and $(t_{27} = 2.65, p < 0.0134)$. Most likely because *Expand* uses a cone-cast to select objects, which results more often in a refinement-step compared to *DrillSample* that casts a ray. No significant difference was measured between *Ray-Casting* and *Drill-Sample*. For the second task, the techniques with an additional refinement step prove to be faster than *Ray-Casting* with *Expand* at $(t_{27} = 7.8545, p < 1.9e-8)$ and *DrillSample* at $(t_{27} = 3.73, p < 0.0009)$, however no significant difference between *DrillSample* and *Expand* could be found. Here, *Ray-Casting* forces the user to successively select and put objects away until the desired object is easily accessible which results in a very time-consuming problem. In task three it took users significantly less time to complete the task when using *DrillSample*, compared to *Ray-Casting* $(t_{27} = 3.24, p < 0.0031)$ or *Expand* $(t_{27} = 2.6, p < 0.0148)$. *Ray-Casting* fails as it did in the second task because both problems force the user to move objects out of view step by step. *Expand* scores much worse than in task two because the targeted object cannot be distinguished out of its spatial context and because *Expand* is only aligning the objects to a two dimensional grid. Between *Ray-Casting* and *Expand* no significant difference could be found.

Significant differences can be seen in Figure 5-2 for the results of the number of selections for task two, three and on average, each with $(F_{2,54} = 10.98, p < 0.0001)$. Task one shows no significant differences at $(F_{2,54} = 0.491, p < 0.615)$ and advises that all selection techniques perform well in the simplest case.

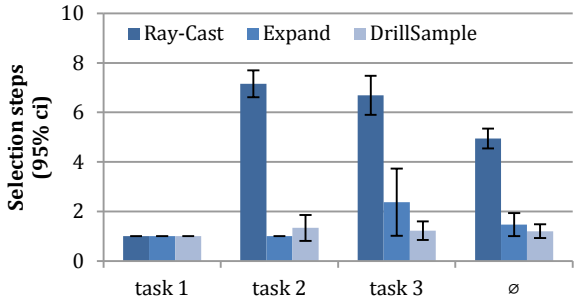

Figure 5-2: Mean Selection steps per task and on average

The pairwise comparison for selection steps on average found the techniques *Expand* ($t_{27} = 15.29, p < 8.04e-15$) and *DrillSample* ($t_{27} = 18.83, p < 4.7e-17$) to be significantly better than *Ray-Casting*, but no significance among another at ($t_{27} = 1.31, p < 0.2$).

Similar to the task completion time, the number of selection steps in the second task were significantly smaller for *Expand* at ($t_{27} = 18.4512, p < 7.78e-17$) as well as for *DrillSample* with ($t_{27} = 13.93, p < 7.55e-14$) compared to *Ray-Casting*, but also *Expand* ($t_{27} = -2.2, p < 0.036$) appears to be slightly less error-prone than *DrillSample*. *Expand* benefits in this scenario from the fact that the targeted object is easily distinguishable, but also from its coarse selection volume where techniques casting a ray may have a hard time to hit an object that is only slightly visible. In task three, likewise for average completion time, we found *DrillSample* having less false selections than *Ray-Casting* ($t_{27} = 16.87, p < 7.29e-16$) and *Expand* ($t_{27} = 2.61, p < 0.0146$). Additionally, *Expand* is significantly better than *Ray-Casting* at ($t_{27} = 8.34, p < 6.01e-9$), too. A possible cause for *Expand* scoring worst in terms of completion time, but not on number of false selections could be that each refinement step costs extra time for the visualization, but also allows users to accidently choose the targeted object each time.

5.2 Subjective Evaluation

Besides the performance measures based on quantitative data, we also examined the user's subjective evaluation on speed and accuracy of each technique. Furthermore, we also include the abstract performance value "ease-of-use" [18] to further evaluate the capabilities of the underlying technique. When answering the questions Q1-Q5, Q7 and Q8, users were able to choose from a 7-point Likert scale. While all questions feature the highest rating at seven, and the lowest at one, Q1 states the best rating with four (appropriate).

The participants found the time allotted for practice appropriate with ($\mu = 3.93$ *and* $\sigma = 0.25$ *at* $\alpha = 0.05$). Using a smartphone to complete the different tasks was rated to be moderately comfortable with ($\mu = 5.72$ *and* $\sigma = 0.98$ *at* $\alpha = 0.05$). As depicted in Figure 5-3 all three techniques were rated at least above average but with significant differences regarding speed ($\chi_2^2 = 10.48, p < 0.0053$), ease of use ($\chi_2^2 = 9.53, p < 0.0085$) and accuracy ($\chi_2^2 = 15.27, p < 0.00048$).

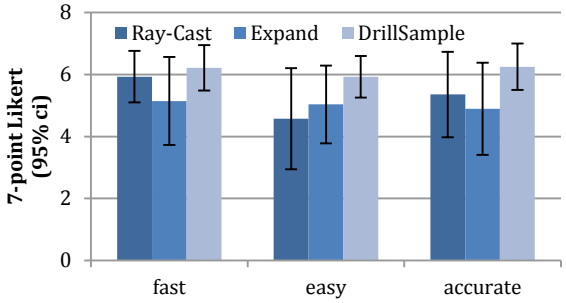

Figure 5-3: Users' average rating of Q3, Q4 and Q5.

Only *DrillSample* was found to be significantly faster than *Expand* in the pairwise comparison ($Z = -2.63, p = 0.0085$). Due to the Bonferroni adjustment, *Ray-Casting* failed to be significantly faster than *Expand* with ($Z = -2.088, p = 0.0368$). *Ray-Casting* was not found to be significantly different from *DrillSample* ($Z = -1.0558, p = 0.29108$). *Expand* was likely rated lower than the other techniques because it triggers refinement too often, while D*rillSample* only asks for refinement if objects overlap. Using *Ray-Casting,* users are not interrupted by a refinement step and might therefore consider it faster. Users' ratings on ease of use found *DrillSample* significantly better than *Ray-Casting* and *Expand* at ($Z = -2.84, p < 0.0045$) and ($Z = 2.91, p < 0.0036$). *Ray-Casting* was insignificantly different to *Expand* with ($Z = -0.89, p = 0.371$) even without the Bonferroni adjustment. Similarly, users found *DrillSample* significantly more accurate than *Ray-Casting* ($Z - 2.69, p < 0.007$) and *Expand* ($Z = -3.17, p < 0.0015$). Likewise *Ray-Casting* showed no significant difference to *Expand* at ($Z = -1.23, p = 0.218$). Both *Ray-Casting* and *Expand* are not easy to use or accurate, if objects are occluded or look very similar. Hence, both factors result in a tedious, and when using *Expand* even a confusing, sequence of interactions to select the desired object.

For question Q6, asking the participant to rank the selection techniques in order of desired use, significant rankings for 1st ($\chi_2^2 = 18.5, p < 9.6e-005$) 2nd ($\chi_2^2 = 12.29, p < 0.0021$) and 3rd ($\chi_2^2 = 9.91, p < 0.007$) could be found as shown in Figure 5-4.

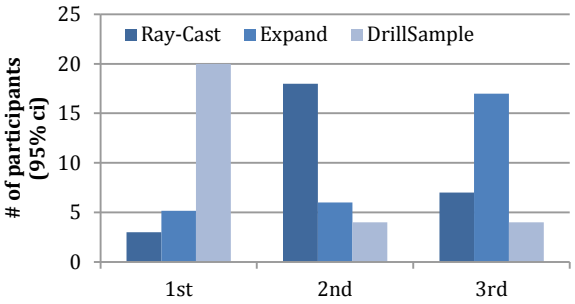

Figure 5-4: Users' rating of Q6.

Rank one was clearly given to *DrillSample* with ($Z = -3.54, p < 0.00039$) and ($Z = -3, p < 0.0027$) significantly outranking *Ray-Casting* and *Expand*. Rank two was given to *Ray-Casting* with ($Z = -2.98, p < 0.0028$) and ($Z = -2.45, p < 0.014$) significantly outranking *DrillSample* and *Expand*. Rank three seems to be given to *Expand*, however it only significantly outranks *DrillSample* with ($Z = -2.83, p < 0.0046$) but not *Ray-Casting* at ($Z = -2.04, p = 0.041$) due to the Bonferroni adjustment. All other pair-wise interaction technique tests show no significant difference.

Users stated all aspects of Q7 evenly important with 6 (important) or higher when answering Q6. Addressing in Q8, how helpful the spatial visualization is, the participants found it useful to very useful with ($\mu = 6.5$ *and* $\sigma = 0.1$ *at* $\alpha = 0.05$).

5.3 Qualitative Evaluation

Based on the 3D formalization principles by [18], we outline a number of factors for the interaction task "3D selection" that influence performance in virtual environments. Since all three evaluated selection techniques are suited or explicitly designed for dense environments, we do not include "density" as a performance factor. The specified factors are:

(1) **Object Size:** This object property is related to the geometric area, a 3D object covers on the output device screen. A selection technique must be capable to select objects of varying size.

(2) **Occlusion:** In any virtual environment, but especially in a dense environment, objects can partially or fully occlude each other which may result in invisible objects. In such

environments, selection must be precise and provide some assisting visualization to identify occluded objects.

(3) **Visual Appearance:** The visual appearance of virtual objects can be of high similarity. Identifying the desired target object can result in problems in dense environments with occluded objects. In such environments, selection must provide an assisting visualization to disambiguate the desired object.

Based on the results from quantitative as well as subjective evaluation, we summarize our findings with respect to the proposed parameters in Table 2.

Table 2: Evaluation of Handheld AR selection techniques with respect to the proposed parameters

	Object Size	Occlusion	Appearance
Ray-Casting	- [1][4]	-	o
Expand	+ [1]	+	-
DrillSample	-	+	+

Previous work [1] [4] report that *Ray-Casting* performs badly for objects covering only a small portion of the screen, while *Expand* performs well for the same case by casting a volume instead of a single ray. Beyond that, our findings indicate that *Ray-Casting* is well suited for selecting non-occluded objects which can be also similar in appearance. However, if the desired object is small and is located amongst similar looking objects, imprecise touch input can evoke wrong selection. Compared to *Ray-Casting*, *Expand* is well suited to select visible or fully occluded objects of varying size. But the grid representation during the refinement step does not provide full spatial correspondence to the original position of the selected objects; hence, precise selection of an object from a set of similar looking objects can be difficult and can result in wrong selections. *DrillSample* also lacks accuracy when selecting small objects due to the underlying use of *Ray-Casting* in combination with the imprecise single touch input. However, since *DrillSample* selects all objects which are cast by the ray, overlapping or occluded objects can be precisely selected due to *DrillSample*'s refinement step. Here, spatial context preservation provides a full overview that allows object disambiguation, which is especially of interest when selecting from a set of similar looking objects.

6. DISCUSSION

We designed the experiment to compare three different techniques in terms of speed, precision and ease-of-use for performing 3D selection tasks with a multi-touch handheld device in a dense AR scene. Many of the outcomes of our performance study were statistically significant which enable us to draw multiple meaningful conclusions. In **H1** we proposed *Ray-Casting* to be best suited for selection of non-occluded objects. Results of completion time for task 1 support H1, since *Ray-Casting* significantly outperforms *Expand*. H1 can further be strengthened by taking the subjective evaluation into account where users considered *Ray-Casting* to be fast. *DrillSample* also performed significantly better than *Expand* for task 1. This indicates the strength of techniques casting a ray instead of casting a cone for visible object selection in close range, since a ray selects fewer objects. Thereby, just a few objects need to be presented at *DrillSample*'s refinement step, while Cone-Casting is always coarser. There, more objects are presented during a refinement step, which takes more time for a user to get an overview before indicating the desired object. Therefore H1 can be supported to be true in terms of speed. Regarding precision, neither performance nor subjective evaluation revealed statistical significance to back up H1. Therefore, we must state H1 to be not true in terms of precision.

Results for evaluating speed and precision, when selecting almost fully occluded objects, clearly reveal *Expand's* and *DrillSample's* strengths. Both perform significantly faster and need less selection steps than *Ray-Casting*, which supports **H2**. Since no significant difference in completion time and interaction steps between *Expand* and *DrillSample* could be found, H2 can be backed up further. These results indicate that *Expand* and *DrillSample* are both equally suited for selecting an occluded object, which highly differs in appearance from the surrounding ones. Regarding precise selection of occluded objects with high visual similarity, *DrillSample* significantly outperforms both baseline techniques in terms of completion time and number of interaction steps. Based on these results, **H3** can clearly be supported. It proves the advantage of our proposed spatial context preservation compared to the grid representation that *Expand* provides. The disadvantage of *Expand*'s detailed visualization becomes even more apparent, since no significant difference in completion time could be found between *Expand* and *Ray-Casting*.

Regarding users' preference, the subjective evaluation clearly reveals users' being in favor of *DrillSample*. It significantly outranked both baseline techniques when users were asked for an overall ranking. This first rank can further be confirmed when looking at the details. Users ranked *DrillSample* highest in terms of speed, precision and ease-of-use. It significantly outperformed Expand in terms of speed, but not Ray-Casting. Since *Ray-Casting* does not provide a refinement step, it tends to be considered fast and "direct". The *DrillSample*'s capability to precisely select the desired object over all three test scenario was ranked significantly best in terms of precision. Finally, the users ranked *DrillSample* significantly best in ease-of-use.

Based on these results and findings, we have developed a set of preliminary guidelines regarding object selection in closer range:

- *Ray-Casting* remains a good alternative selection technique, as long as objects are fully visible.
- *Expand* remains a good alternative for visible or occluded objects of varying object size, as long as they differ in visual appearance.
- For visible or occluded objects, independent of their visual appearance, *DrillSample* is the best general purpose method.

7. CONCLUSION & FUTURE WORK

In this work we explore 3D selection techniques in handheld AR environments by evaluating three techniques. Precise selection of objects in dense one-handed handheld AR is our main motivation. Therefore, we intended to reduce multi-touch input due to implicit restrictions having only one hand available for selection. Our approach requires only single touches as input and splits up the procedure into two steps. We preserve the spatial context if multiple objects have been indicated as targets, to allow for disambiguation and precise selection of occluded objects or objects with high similarity in visual appearance.

The performance study clearly revealed the strengths of the *DrillSample* technique compared to related work in precise selection of objects in dense environments within close range.

Although we tested the new technique in handheld AR, the technique applies for dense VEs as well. As future work we will explore various object density and distance combinations. We plan to investigate using *DrillSample* with Cone-Casting to provide accurate selection of smaller objects at a larger distance. Furthermore, we want to examine performance and usability of *DrillSample* when selecting objects of varying size. Therefore, we will take a closer look on competing techniques like SQUAD [2], Expand [1] and Dual-Finger Selection Techniques [14] for latter evaluation.

8. REFERENCES

[1] J. Cashion, C. Wingrave, and J. J. LaViola, "Dense and dynamic 3D selection for game-based virtual environments.," in *IEEE Virtual Reality*, 2012, vol. 18, no. 4, pp. 634–42.

[2] R. Kopper, F. Bacim, and D. a. Bowman, "Rapid and accurate 3D selection by progressive refinement," in *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, 2011, pp. 67–74.

[3] A. Mossel, C. Schönauer, G. Gerstweiler, and H. Kaufmann, "ARTiFICe-Augmented Reality Framework for Distributed Collaboration," *To appear in International Journal of Virtual Reality*, 2013.

[4] D. Bowman, E. Kruijff, J. J. LaViola Jr., and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2005.

[5] I. Poupyrev and M. Billinghurst, "The go-go interaction technique: non-linear mapping for direct manipulation in VR," in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, 1996, pp. 79–80.

[6] J. Liang and M. Green, "JDCAD: a highly interactive 3D modeling system," in *Proceedings of Third International Conference on CAD and Computer Graphics*, 1994, pp. 217–222.

[7] A. Forsberg, K. Herndon, and R. Zeleznik, "Aperture based selection for immersive virtual environments," in *Proceedings of the 9th ACM symposium on user interface software & technology*, 1996, pp. 95–96.

[8] J. S. Pierce, A. Forsberg, M. J. Conway, S. Hong, R. Zeleznik, and M. Mine, "Image Plane Interaction Techniques in 3D Immersive Environments," in *Proceedings of the 1997 Symposium on Interactive 3D graphics (I3D '97)*, 1997, pp. 39–43.

[9] D. A. Bowman and L. F. Hodges, "An Evaluation of Techniques for Grabbing and Manipulating Objects in Immersive Virtual Environments Arm-Extension Ray-Casting," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 35–38.

[10] G. Lee, U. Yang, Y. Kim, D. Jo, and K. Kim, "Freeze-Set-Go interaction method for handheld mobile augmented reality environments," in *VRST '09 Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, 2009, pp. 143–146.

[11] W. Hürst and C. Van Wezel, "Multimodal interaction concepts for mobile augmented reality applications," *Advances in Multimedia Modeling*, pp. 157–167, 2011.

[12] W. Hürst and C. Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," in *Multimedia Tools and Applications*, 2012.

[13] H. Benko, A. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 1263–1272.

[14] C. Telkenaroglu and T. Capin, "Dual-Finger 3D Interaction Techniques for mobile devices," *Personal and Ubiquitous Computing*, Sep. 2012.

[15] Qualcomm, "Vuforia," 2012. [Online]. Available: http://www.qualcomm.com/solutions/augmented-reality.

[16] Unity Technologies, "Unity3D," 2012. [Online]. Available: http://www.unity3d.com/.

[17] M. E. Mündel, *Motion and Time Study: Improving Productivity*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1978.

[18] D. a. Bowman and L. F. Hodges, "Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments," *Journal of Visual Languages & Computing*, vol. 10, no. 1, pp. 37–53, Feb. 1999.