Bubble Radar: Efficient Pen-Based Interaction

Dzmitry Aliakseyeu[1] Miguel A. Nacenta[2] Sriram Subramanian[2] and Carl Gutwin[2]

[1]Faculty of Industrial Design
Eindhoven University of Technology
Den Dolech 2, Eindhoven, The Netherlands
d.aliakseyeu@tue.nl

[2]Department of Computer Science
University of Saskatchewan
110 Science Place, Saskatoon, SK, Canada
{nacenta, sriram, gutwin}@cs.usask.ca

ABSTRACT
The rapid increase in display sizes and resolutions has led to the re-emergence of many pen-based interaction systems like tabletop and wall display environments. Pointing in these environments is an important task, but techniques have not exploited the manipulation of control and display parameters to the extent seen in desktop environments. We have overcome these in the design of a new pen-based interaction technique – Bubble Radar. Bubble Radar allows users to reach both specific targets and empty space, and supports dynamic switching between selecting and placing. The technique is based on combining the benefits of a successful pen-based pointing technique, the Radar View, with a successful desktop object pointing technique – the Bubble Cursor. We tested the new technique in a user study and found that it was significantly faster than existing techniques, both for overall pointing and for targeting specific objects.

Categories and Subject Descriptors
H5.2 **[Information interfaces and presentation]**: User Interfaces. - Graphical user interfaces

General Terms
Performance, Design, Experimentation, Human Factors.

Keywords
Interaction techniques, object pointing, reaching, large-display systems, multi-display systems.

1. INTRODUCTION
Rapid increases in display sizes and screen resolution have seen greater research attention paid to desktop pointing techniques. Researchers have proposed many new pointing techniques, such as Area Cursor [23], Object Pointing [10], Semantic Pointing [4] and Bubble Cursor [9]. These techniques improve desktop pointing by exploiting the fact that virtual pointing can surpass physical pointing by manipulating the control and display parameters. These object pointing techniques work with the standard mouse and have been shown to improve the selection of discrete targets in the desktop environment.

The emergence of new display environments such as tabletops [7] and wall displays [22]has led to interaction techniques that are moving away from keyboard-and-mouse style of interaction, and many of them use pen-based [20] or finger-based [24] input devices. Pointing is still an important part of these environments. Most pointing techniques (like pick-and-drop [20]) couple the motor and visual spaces (because the pen/finger is physically placed on the object of interest; there is no cursor). This coupling of the motor and visual spaces, however, limits the user's virtual pointing ability to their physical pointing ability.

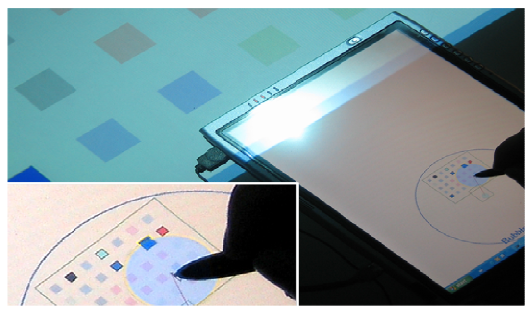

Figure 1. Bubble Radar. The inset picture shows the user selecting a target in a reduced representation map that uses Bubble Cursor - an enhanced area cursor.

Recently, many pen-based interaction techniques like Radar Views[18], Drag-and-Pop[2] and Pantograph [11], that decouple the motor and visual spaces, have been proposed. This decoupling allows manipulation of the control and display parameters to improve pointing in these environments. However, pen-based techniques have not yet harnessed the benefit demonstrated by mouse-and-cursor-based pointing innovations; our goal in this project is to determine what potential improvements these techniques can bring to pen-based interactions.

Applying object pointing techniques to pen-based situations presents several problems. First, many object pointing techniques only allow users to select specific targets, so the empty space between targets is not available or is more difficult to access. The second limitation of the object pointing techniques is that they cannot be applied directly to techniques like pick-and-drop, which rely on the coupling of the motor and visual spaces.

We have designed a new pen-based technique that overcomes these limitations. We enable access to empty space by allowing users to switch modes between object pointing and regular pointing, and we modify the object approach so that it can be applied to existing pen-based techniques. Our new technique – Bubble Radar – allows users to reach both specific targets and empty space, and supports dynamic switching between selecting and placing. The technique is based on combining the benefits of a successful pen-based pointing technique, the Radar View, with a successful desktop object pointing technique – the Bubble Cursor.

We tested Bubble Radar in a user study where we compared the technique with four interaction techniques. We used three existing techniques (Radar View, Pick-and-Drop, and Push-and-Pop) and a fourth that is a second hybrid, Bubble Pick-and-Drop. We found that the Bubble Radar was significantly faster than the other techniques when considering both specific targets and empty space.

The paper makes the following contributions

- We show that the bubble idea can be applied to different techniques such as Radar.
- We show that bubble does not work for all pointing techniques, e.g. Pick-and-Drop.
- We show that Bubble Radar is better than previous best techniques Push-and-Pop, Radar, and Pick-and-Drop.
- We show that mode switch can be added to bubble.

The rest of the paper is organized as follows: first we describe related work, and then present the new interaction technique. Then, we report on the experiment and discuss the implications of our results for design.

2. RELATED WORK

All material on each page should fit within a rectangle of 18 x A number of pointing interaction techniques have been developed for placing objects in environments like desktops, large walls, tabletops and multi-display settings. Most of the desktop pointing techniques use the mouse as the input device, while wall and multi-display environments usually use pens (or finger pointing) as the primary input. Below we review the important mouse and pen-based interaction techniques for pointing.

2.1 Pointing Techniques for Mouse-Based Devices

Balakrishnan [1] surveys several pointing techniques that have been proposed within the context of desktop pointing. Area Cursors [23], Object Pointing [10], Bubble Cursor [9], and Semantic Pointing [4] are all examples of mouse-based desktop pointing techniques that allow users to target specific objects.

Area Cursor: In Area Cursor [23] the cursor has a larger activation area than normal "single point" cursors. To over-come occlusion problems and problems with multiple tar-gets within the same area, Worden et al. [23] created an area cursor with two hotspots – one is a single point in the center and another is the whole cursor. When the cursor touches two or more targets, the single point hotspot activates; in other situations, the whole area cursor is active.

Bubble Cursor: The Bubble cursor [9] is an improvement of area cursors that allows selection of discrete target points by using Voronoi regions to associate empty space with nearby targets using hotspots. An advantage of this technique is that even in relatively high target-density workspaces, users can benefit from improved targeting times and reduced error rates.

Object Pointing: In Object Pointing [10] the size of targets is maintained while the size of the empty space between tar-gets ("gaps") is completely removed, so the cursor "jumps" between targets.

Semantic Pointing: In Semantic Pointing [4] the empty spaces between targets are not completely removed, but instead made smaller in motor space by dynamically adjusting the control-to-display ratio.

Although it was shown that each of the above techniques outperforms traditional pointing, little is know how they are compared with each other [1]. From the above mentioned techniques only Bubble Cursor – Object Pointing pair was compared. The results of this comparison showed the superiority of Bubble Cursor [9].

2.2 Pen-Based Pointing Techniques

A number of interaction techniques have also been proposed to improve pen-based interaction in multi-display and large-display environments. Some of these techniques only support object selection, while others allow both object selection and free-space aiming. Henceforth, by pen-based interaction we mean techniques that use a pen or a finger like pointing device as input.

Pick-and-Drop [20] is one of the first techniques proposed for multi-display reaching. Here the user can 'pick up' an object by touching it with a digital pen (or any other suit-able device), and then 'drop' the object anywhere in the workspace by repeating the touch action in the desired location. Stitching [12, 13] is another variation on pick-and-drop that allows ad-hoc and dynamic creation of multi-display environments.

Drag-and-Pop [2] is a technique introduced primarily for selection of specific targets in large wall displays. When the user starts moving in a certain direction, proxies of all potential targets in that direction are brought closer to the user. It is also possible for the user to ignore these proxies and continue moving the object, in which case the proxies are dismissed and the user performs a familiar drag-and-drop action to place the object in an empty space. However, placing objects can be arduous, especially when trying to place them in distant corners of the large display, as the user does not get any kind of system support when placing objects in empty spaces. Bezerianos and Balakrishnan [3] introduced 'vacuum filtering' as a lightweight extension to Drag-and-Pop that provides users with more control over the number of proxies created.

Movement amplification techniques are another style of interaction based on the idea of transporting the user's cursor to the whereabouts of the target. Techniques like Throw [8, 24, 18, 11], Pantograph [11, 18], and Flick [17] all amplify user movements, and avoid the limitations of reaching distant corners of large displays that is typical of Pick-and-Drop and Drag-and-Pop.

TractorBeam [19] is a hybrid point-touch technique that allows users to reach distant objects on tabletop displays. Using a 6DOF stylus, the user points at the display, and a cursor appears on the display to show the current trajectory of the stylus. To improve object acquisition time, TractorBeam can provide a selection aid through expanding the cursor, expanding the target or snapping to the target.

Push-and-Pop [6] combines Drag-and-Pop with Push-and-Throw. Push-and-Pop has a takeoff area that semantically represents the whole display. All potential targets are brought into this takeoff area, irrespective of the movement direction. Position of these icons mirrors the actual positioning of the icons on the display. In continuous mode, Push-and-Pop becomes a movement amplification technique (Push-and-Throw [11, 6]). Users can switch into continuous mode by moving the pointer back into the initial movement region.

RadarView. The Radar technique uses a reduced representation (a map) of the entire environment. When the pen touches the object the map appears, placed so that the position of the pen is the same in both representations. The Radar map is different from that of Push-and-Pop [6]: the Radar map proportionally reduces both the size of objects and space between them and allows continuous positioning of the object within the map.

Nacenta et al. [18] compared several pointing techniques like the Radar View, Pantograph and Pick-and-drop. They found that the Radar View was faster than movement amplification techniques like Pantograph and Slingshot. In a follow-up study, Collomb et al. [6] found similar results – that bringing objects closer to the user improved pointing times compared to moving the cursor to the target's location.

3. BUBBLE RADAR

Since Bubble Cursor has been shown to be the fastest desktop pointing technique, we decided to adapt it to pen-based pointing. In designing the Bubble Radar, we had to address the two main limitations of the Bubble Cursor.

A first limitation of bubble cursor is that it only allows users to target specific objects. Bubble cursor is a circular semi-transparent area cursor with a cross-hair at the center of the bubble. A simple algorithm continuously updates the radius of the bubble cursor so that there is always one and only one target within its hotspot. The algorithm also ensures that the target within its hotspot is the closest target to the cross-hair.

Second, the bubble cursor has only been shown to work with the mouse. It benefits from the decoupling of the motor and visual spaces, by allowing the user to primarily focus on the visual space and react to changing hotspot radius without being disturbed by disproportionate mouse movement and the increase in hotspot radius.

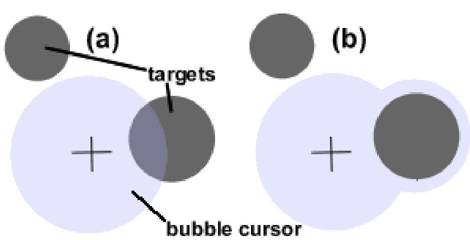

Figure 2. Bubble cursor. (a) cursor changes its size dynamically such that only the target closest to the cursor centre is selected. (b) The bubble cursor morphs to encompass a target when the basic circular cursor cannot completely do so without intersecting a neighboring target [9].

However, the Radar technique successfully decouples visual and motor space by creating a reduced-representation map. This allows us to combine the Radar with the bubble cursor to design Bubble Radar. With Bubble Radar, when the mode is for specific targets, the pen tip changes to an area cursor with a dynamically changing activation area that selects only one object at a time (as in bubble cursor, Figure 3(4)). When the mode is for empty space, the pen tip changes to a cross-hair so that the user can continuously position the object within the map (as in Radar, Figure 3(5)). Users are free to go back and forth between two modes any number of times before performing a pointing task. Figure 3 shows how Bubble Radar works in each of the modes.

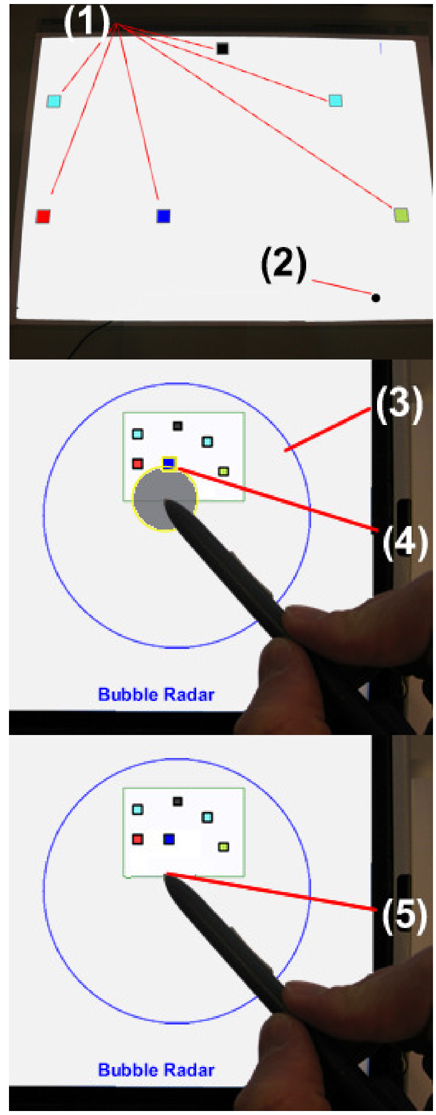

Figure 3: Bubble Radar. (1) potential targets; (2) object to position; (3) reduced representation of environment within the Bubble Radar; (4) target closest to the bubble cursor is selected if pen will be released now the object will be transferred to that target; (5) in simple radar mode the object can be dropped at any place.

The mode-switch button is activated by the non-dominant hand. This choice of mode switching is based on the results of a comparative study conducted by Li et al [14]. In their study, the authors show that mode switching with the non-dominant hand is faster than most other means of mode switching such as using pen pressure, buttons on the pen, or press-and-hold techniques.

4. BUBBLE PICK-AND-DROP

We also designed a second technique, Bubble Pick-and-Drop. This technique combines the bubble cursor with pick-and-drop. When initiating the technique the bubble is displayed, selecting the closest target. The bubble, however, is only displayed when the pen is in proximity of the table. Thus if the user moves the pen well above the display they only get visual aid at the end of the movement. We realize that Bubble Pick-and-Drop has several limitations when compared to Bubble Radar. The main purpose of designing this technique was to see if Bubble cursor universally improves targeting times in all pen based pointing techniques despite our observation that bubble cursor needs a decoupled motor and visual space to be effective.

5. USER STUDY

We conducted a user study to compare five interaction techniques, all of which allow both targeting specific objects and aiming at free space. We compare our two new combination techniques with three existing techniques – Radar, Pick-and-Drop and Push-and-Pop.

5.1 Apparatus

A tabletop system was set up using a desktop PC and a pen activated tablet PC (Size: 15.5 x 21cm, Resolution: 768x1024 pixels). The PC controlled a top-down projector projecting an image of size 75 x 50 cm (1024x768 pixels) on a large white table (100x80cm) that incorporated a pen-driven digitizer (Wacom UltrapadA2) also connected to the PC. In our experiments, a tablet PC was fixed into position close to the subject to allow higher display resolution for user input (see figure 4). We used a tablet PC and a digitizer to create a large single display with 2 input regions. We needed this to allow us to do the experiment in a setting that is larger than our digitizer. Radar, Bubble Radar, and Push-and-Pop do not require pen movement between input regions; Pick-and-Drop and Bubble Pick-and-Drop required movement between regions. The tablet was large enough to allow complete Push-and-Pop actions. The same pen was used on both regions.

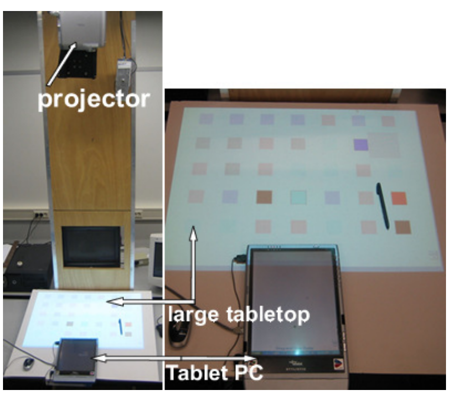

Figure 4: Experimental setup

5.2 The Projected Space and Users Task

As was discussed earlier, the goal of this study was to compare techniques based on two types of activity: targeting specific objects (potential objects are known to the system), and aiming at free space (the potential target is an empty space or unknown to the system).

In order to measure the performance in these two activities we varied the task in a way that forced participants to switch between targeting specific objects and aiming at free space.

Each subject was instructed to move several objects from the tablet PC to various targets on the table, using different interaction techniques. The object to be moved was a circle (5mm diameter), and the targets were colored square icons of varying sizes placed in the projected space of the table. The position of the targeted icon was randomly selected at 6 possible positions: 2 distances of 30 and 60 cm and three different angles of -25^0, 0^0 and $+25^0$ (see Figure 5).

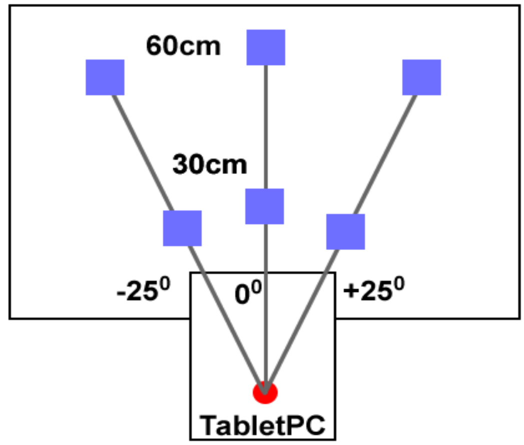

Figure 5: Target positions

For targeting specific objects the user had to match the object color to the appropriate target color. There was always only one matching target. In addition to the target icon the projected space had also included 5 potential targets and 29 differently colored icons that serve as distracters (see Figures 6 and 7). Distracters were made semi-transparent so increase visibility of potential targets. Potential targets act as distracters when targeting specific objects while the colored icons and the potential targets act as distracters when aiming at free space. The positions of potential targets and distracter icons were randomly generated in every trial.

For aiming at free space we indicated to the user a big target which was shaded in grey and appeared behind one of the other smaller targets (see Figure 6). This big target acted as background and was only visible when the participant had to perform aiming in free space. These targets were large enough with index of difficulty between 2 and 3 to simulate aiming in free space.

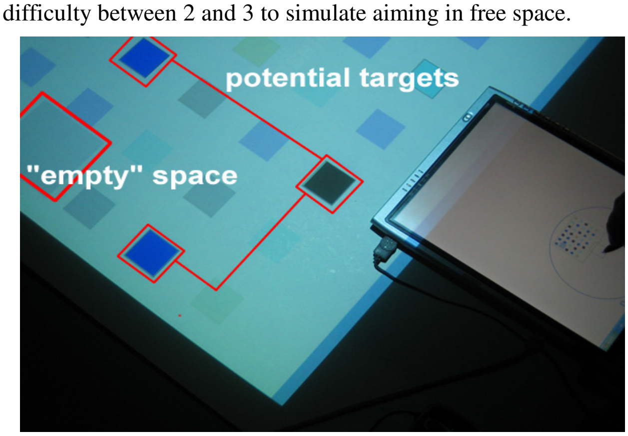

Figure 6: Placing with Radar View

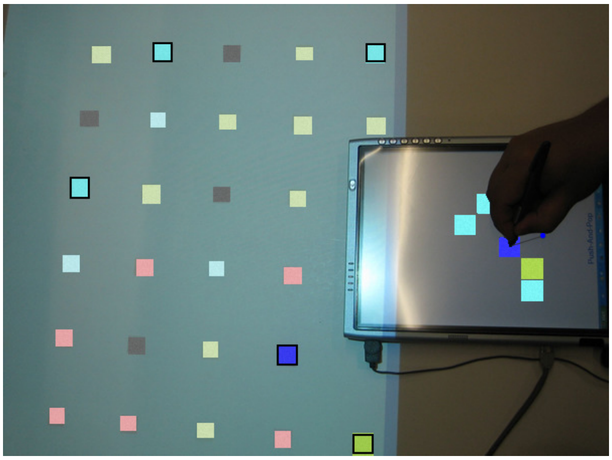

Figure 7. Selecting with push-and-pop

The trial was considered a 'hit' if the center of the object ended up inside the area of the correct target, and an error otherwise. There are three possible errors: A drop, a miss and a false hit. A drop occurs when the user drops the object close to the starting position (possibly because of loosing control of the input device). A miss occurs when the user drops the object in a location where there is no target. A false hit occurs when the user drops the object in the wrong target. The system provided distinctive sound feedback for hits and misses. The subjects were asked to perform repetitions of this task using the different interaction techniques, as fast as possible but without missing the target.

5.3 Design

The experiment was conducted with 8 right-handed subjects (1 female and 7 males) between the ages of 18 and 31. All subjects had previous experience with graphical interfaces and all were tested individually. The experiment used a 5x2x3 within-participants factorial design with a variety of planned comparisons. The factors were:

Technique (Pick-and-Drop, Bubble Radar, Radar View, Bubble Pick-and-Drop, Push-and-Pop)

Target distance (30cm, 60cm)

Target size (square specific objects with side of 43 mm, 20 mm for 30cm; 40, 19 mm for 60cm; square empty space objects with side 90 mm for both distances)

For each technique, size, and distance, participants completed three training trials and twelve test trials, for a total of 90 training trials and 360 test trials. The distance and target sizes for targeting specific objects were chosen to allow comparison of techniques for index of difficulty (3, 4 and 5). Index of difficulty was calculated using the equation $ID=\log_2(D/W+1)$ [15]. The index of difficulty when aiming at free space was 2.1 and 2.9 for the different distances.

At the end of the experiment participants were also asked to complete a questionnaire to rank the different techniques in order of preference. Trial completion times and error rate (number of misses) were used as the main measures to compare the different techniques.

5.4 Interaction Techniques

The five interaction techniques compared in the study were Bubble Radar, Bubble Pick-and-Drop, Radar Views, Pick-and-Drop and Push-and-Pop. The implementation of Bubble Radar and Bubble Pick-and-Drop was as described in the previous sections.

The implementation of Pick-and-Drop was similar to the original implementation described by Rekimoto [20], and behaved as outlined earlier. The implementation of the Radar View was similar to the implementation described in [18] (also discussed earlier). The mapping coefficient (the reduction ratio) was 15 for both the Radar and the Bubble Radar.

The implementation of Push-and-Pop was adapted from the original implementation [6]. The size of big targets had to be reduced to fit within the takeoff area. This however, did not change the ID of the task since the reduction of size also led to an equivalent reduction of the distance. The continuous part of the technique (push-and-throw) was implemented without acceleration [6, 11], since the distances were relatively small. The mapping coefficient was also 15.

5.5 Results

We used three performance measures to evaluate the different interaction techniques – mean completion times, error rates and subjective preference scores.

5.6 Time

Overall Performance: The overall mean completion times across all conditions was 1.39 seconds (standard deviation = 0.9 seconds). One-way repeated-measures ANOVA showed that interaction technique had a significant effect on the trial completion time $F_{4,35}=7.985$, $p<0.001$)

Targeting Specific Objects: The mean completion time for selecting specific targets was 1.077 seconds (standard deviation = 0.566 seconds). One-way repeated-measures ANOVA showed that the interaction technique had a significant effect on the trial completion time for targeting specific objects ($F_{4,28}=40.965$, $p<0.001$).

Post-hoc pair-wise comparisons (Tamhane, unequal variances) showed that all pairs except (Radar, Bubble-Pick-and-Drop), (Radar, Pick-and-Drop), (Pick-and-Drop, Bubble Pick-and-Drop) were significantly different at $p<0.01$.

Figure 8 show the mean trial completion times with standard error for all interaction techniques in this experiment. As can be seen from figure, Bubble Radar was the fastest technique, followed by Push-and-Pop, Radar, Bubble Pick-and-Drop and Pick-and-Drop.

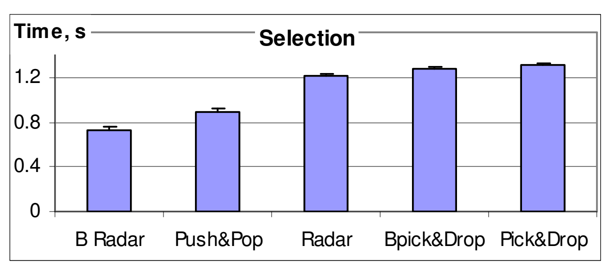

Figure 8. Mean trial completion times with standard error for interaction techniques when selecting specific targets.

Aiming at Free Space: The mean completion time for aiming at empty space was 1.392 seconds (standard deviation = 0.918 seconds). One-way repeated-measures ANOVA showed that the interaction technique had a significant effect on the trial completion time for aiming at empty space ($F_{4,28}=31.490$, p<0.001).

Post-hoc pair-wise comparisons (Tamhane, unequal variances) showed that all pairs except (Radar, Bubble Radar) and (Pick-and-Drop, Bubble Pick-and-Drop) were significant at p<0.01.

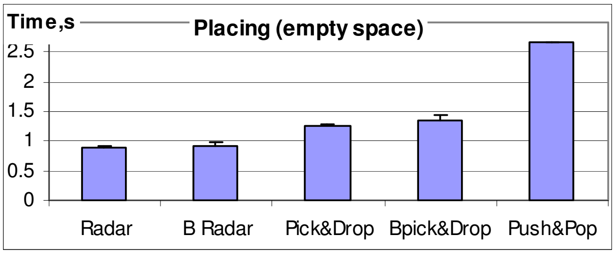

Figure 9. Mean trial completion times with standard error for interaction techniques when aiming at free space.

As can be seen in Figure 9, Bubble Radar and Radar were the fastest two techniques, followed by Pick-and-Drop, Bubble Pick-and-Drop and Push-and-Pop.

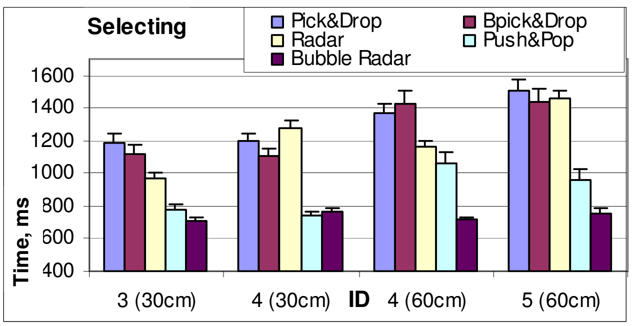

Figure 10. Mean trial completion time with standard error for targeting specific objects.

Index-of-Difficulty Performance: We performed a separate comparison of the interaction techniques for the different indices of difficulty (ID) used in the experiment. These comparisons showed that for each ID, interaction technique had a significant effect on trial completion time. At ID = 3, $F_{4,28}$=7.6 (p<0.05); ID=4: $F_{4,28}$=41.4 (p<0.001); ID=5: $F_{4,28}$=21.1 (p<0.01). Figure 10 shows the mean completion times for the different interaction techniques for each ID.

Target Position: The comparison based on the targets positions (angle -25^0, 0^0, +25^0) showed no significant differences for either selecting ($F_{2,14}$=0.56, p=0.946) or placing ($F_{2,14}$=0.344, p=0.713).

5.7 Error Ratess

Drop Error Rate: The number of drops was 1 for Radar, Pick-and-Drop and Bubble Pick-and-Drop, 3 errors for Bubble Radar and 8 for Push-and-Pop.

False Hit Error Rate: There were no false hits with Radar and Pick-and-Drop, 2 false hits with push-and-pop 8 with Bubble Radar and 28 with Bubble Pick-and-Drop.

Miss Error Rate: The number of misses were 4 for Bubble Radar, 5 for Bubble Pick-and-Drop, 21 for Pick-and-Drop, 24 for Push-and-Pop and 26 for Radar.

5.8 Subjective Preference

At the end of the experiment participants were asked to rank each technique based on perceived control, tiredness, speed and overall preference. Each technique was assigned a number from 1 to 5 with 1 being best and 5 being worst. Bubble Radar was the most preferred technique while Pick-and-Drop was least preferred. Figure 11 shows the mean value for the ranking of each technique.

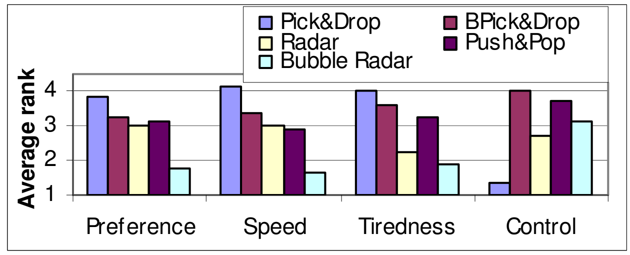

Figure 11. Mean values for the ranking of each technique.

6. DISCUSSION

In the sections below we discuss several issues raised by the study: differences between the Bubble Radar and the ordinary Radar caused by the addition of the bubble; differences between the two versions of Pick-and-Drop; differences between the two Bubble-based techniques and Push-and-Pop; and our overall conclusions about the effectiveness of the bubble addition and the necessity for a mode switch in the combined techniques.

6.1 Bubble Radar vs. Radar

Bubble Radar was at least twice as fast as the Radar View in targeting specific objects (see Figure 11). This difference grows with increasing index of difficulty, since the performance of Bubble Radar is not affected by the target size. Even in situations where there are many potential targets, or there is little empty space between targets, the performance of the Bubble Radar never becomes worse than Radar. In the extreme case where there is no empty space, the two techniques will become equal. A study conducted by Ren and Moriya [21] showed that the minimum size of a target that can be acquired by a pen is 1.8mm. Bubble Radar removes this limitation, allowing designers to conserve screen real estate when needed.

In the case of aiming at free space, Bubble Radar was nearly as good as Radar, and we found no significant difference between the two techniques. Of course, the only real difference between the two techniques in this case is the additional mode-switch button that is part of Bubble Radar, but as Li et al. [14] showed, this adds less than 150 ms to the overall task completion times. In terms of total errors, Bubble Radar was better than Radar (15 errors for Bubble Radar versus 27 for Radar).

6.2 Bubble Pick-and-Drop vs. Pick-and-Drop

The main reason we included Bubble Pick-and-Drop in our study was to see if it improved on Pick-and-Drop. We found that there was no significant difference between these two techniques, either for targeting specific objects or for aiming at free space. As explained earlier, Bubble Pick-and-Drop has its limitations – users get no visual feedback when they move the pen beyond the hover zone of the touch-sensitive surface. This lack of feedback created some delay in recognizing if the Bubble was actually touching the desired target. We noticed that subjects who tried to estimate the Voronoi regions without waiting for the bubble visualization were generally faster. One way to overcome this limitation would be to display the Voronoi regions on the workspace. Further, the mode switch was implemented as a separate button controlled by the user's non-dominant hand. This can create problems if the user has to reach distant target that requires walking up to it. Any correction in the mode means the user has to carry this button or more likely return to the initial position to change modes.

This result suggests that Bubble cursor has to be added judiciously to pen based techniques since it does not guarantee performance gains.

6.3 Bubble-modified techniques vs. Push-and-Pop

The overall analysis of the study showed that Bubble Radar is significantly faster then Push-and-Pop. However, the individual tests for different indices of difficulty did not show any differences, at least for selecting.

The main difference between the Push-and-Pop and Bubble Radar (in selection mode) is the size of the potential targets. Push-and-Pop does not reduce the size of targets, while the Radar reduces them based on the mapping coefficient. Both approaches have their drawbacks. For Bubble Radar, if the mapping coefficient is too large, then targets will become too small to identify; if the mapping coefficient is too small, not all targets will fit in the map. In the case of Push-and-Pop, if the targets are too big or the number of potential targets is too large, then the targets might not fit the working (visible) area. In addition, the number of potential targets that are between all possible targets also heavily affects the technique's performance.

Push-and-Pop was slowest when aiming at free space. This is partly because of the mode-switching mechanism in this technique, and partly because of the placing strategy. The mode switch has the advantage that it does not need any additional buttons; however, it is two to three times slower than the separate-button method. In addition the Push-and-Pop mode switch requires that the user move back to the starting position, while in Bubble Radar the user can switch modes in any position.

Push-and-Pop's placing technique uses dynamic acceleration to improve upon a basic pantograph. We did not use any acceleration in our implementation of Push-and-Pop; however, as noted by the authors [6], the acceleration only helps the acquisition of smaller targets. In our experiment the size of empty space was relatively large (ID of 2.1 and 2.9), and it is unlikely that acceleration would improve performance.

6.4 Effect of the bubble

In general, bubble-enhanced techniques have a number of advantages over traditional pointing methods. First, a mode-switching bubble scheme can successfully be combined with most existing pen-based techniques that decouple visual and motor spaces. This brings the benefits of object pointing to a variety of new techniques. Second, the bubble-based techniques allow overshooting of targets that are located on the edge of a cluster. This makes the target depth nearly infinite, allowing users to acquire it very quickly.

6.5 Effect of Mode Switching

In our implementation of Bubble Radar we used a mode switch button for activation, using the non-dominant hand. This might not be suitable for all applications, especially when the non-dominant hand is used for other activities like gesturing or tool control. Other mode-switch strategies are available that do not require two-handed input.

Mode-switching is, however, an integral part of adding the bubble to another technique. If mode-switching is not well implemented, than any benefit derived from using the bubble might disappear, depending on how often targeting specific objects is performed when compared to aiming at free space. If a user is more likely to perform a free-space aiming task than a targeting task, then the mode-switch can be reversed – that is, used to activate the targeting task rather than the aiming task.

7. CONCLUSION

In this paper we presented Bubble Radar, a interaction technique for pen-based systems. We designed Bubble Radar by combining the benefits of bubble cursor with radar views, while removing the limitations of each technique. Through a user study we found that Bubble Radar is significantly faster and more accurate than traditional methods. Bubble Radar allows designers to save screen real estate in pen-based devices by enabling users to target objects that are smaller than 1.8 mm (the minimum target that can be acquired by a pen).

Our studies also show that combining bubble cursor with pen based interaction techniques does not always result in improvement. We found that bubble pick-and-drop did not perform any better than pick-and-drop. Our studies also show that using a mode switch with the original bubble cursor to allow selecting empty space and specific targets can effectively remove the limitations of the original technique.

In the future we plan to explore the role of such reaching techniques in collaborative settings. We are particularly interested in ways to provide appropriate feedback for maintaining awareness of people's actions on the tabletop. We are also investigating ways to use our new techniques for handoff actions (from a sender to a receiver) where maintaining awareness of each others' actions is important.

8. REFERENCES

[1] Balakrishnan, R.. Beating Fitts' law: virtual enhancements for pointing facilitation, Int. J. Human-Computer Studies 61(2004), 857–874.

[2] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. Drag-and-pop and Drag-and-pick: Techniques for accessing remote screen content on touch- and pen operated systems. Proc. of Interact 2003, (2003), 57-64.

[3] Bezerianos, A., and Balakrishnan, R. The Vacuum: Facilitating the manipulation of distant objects. Proc. of CHI 2005, ACM Press (2005), 361-370.

[4] Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. Proc. of CHI 2004, ACM Press (2004), 519-525.

[5] Cockburn, A. and Firth, A. Improving the acquisition of small targets. Proc. of British HCI Conference. (2003), 181-196.

[6] Collomb, M., Hascoët, M., Baudisch, P., and Lee, B. Improving drag-and-drop on wall-size displays. Proc. of GI 2005, (2005), 25-32.

[7] Dietz, P., Leigh, D., DiamondTouch: A Multi-User Touch Technology. Proc. of UIST 2001, ACM Press (2001), 219-226.

[8] Geiβler, J. Shuffle, throw or take it! Working Efficiently with an Interactive Wall. Proc. of CHI 1999, ACM Press (1998), 265-266.

[9] Grossman, T., and Balakrishnan, R. The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. Proc. of CHI 2005, ACM Press (2005), 281-290.

[10] Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in GUIs. Graphics Interface, (2004), 9-16.

[11] Hascoët, M. Throwing models for large displays. Proc. of HCI2003, British HCI Group (2003), 73-77.

[12] Hinckley, K., Synchronous Gestures for Multiple Users and Computers. Proc. of UIST 2003, ACM Press (2003), 149-158.

[13] Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: pen gestures that span multiple displays. Proc. of AVI '04, ACM Press (2004) 23-31.

[14] Li,Y., Hinckley K., Guan, Z., Landay J. Experimental Analysis of mode switching techniques in pen-based user interfaces. Proc. of CHI 2005, ACM Press (2005), 461-470

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

[16] McGuffin, M., and Balakrishnan, R.. Acquisition of expanding targets. Proc. of CHI 2002, ACM Press (2002), 57-64.

[17] Moyle, M., Cockburn, A. Analyzing Mouse and Pen Flick Gestures. Proc. of SIGCHI-NZ, (2002), 266-267.

[18] Nacenta, M.A., Aliakseyeu, D., Subramanian, S., and Gutwin, C. A comparison of techniques for Multi-Display Reaching. Proc. of CHI 2002, ACM Press (2002), 371-380.

[19] Parker, K., Mandryk, R., Nunes, M., Inkpen, K. TractorBeam Selection Aids: Improving Target Acquisition for Pointing Input on Tabletop Displays. Proc. of Interact 2005, Springer (2005), 80-93.

[20] Rekimoto, J. Pick-and-Drop A Direct Manipulation Technique for Multiple Computer Environments. Proc. of UIST 1997, ACM Press (1997), 31-39.

[21] Ren, X, Moriya, S. Improving Selection Performance on Pen-Based Systems: A Study of Pen-Based Interaction for Selection Tasks. ACM Transactions on Computer-Human Interaction, Vol. 7, No. 3, (2000), ACM Press, 384-416.

[22] Streitz, N., Tandler, P., Muller-Tomfelde, C., Konomi, S. Roomware: Towards the Next Generation of Human-Computer Interaction based on an Integrated Design of Real and Virtual Worlds. In: J. Carroll (Ed.), Human-Computer Interaction in the New Millennium, Addison-Wesley (2001), 553-578.

[23] Worden, A., Walker, N., Bharat, K., and Hudson, S. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. Proc. of CHI 1997, ACM Press (1997), 266-271.

[24] Wu, M., and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Proc. of UIST 2003, ACM Press (2003), 193-202.