AD-Binning: Leveraging Around-Device Space for Storing, Browsing and Retrieving Mobile Device Content

Khalad Hasan[1], David Ahlström[2], Pourang Irani[1]

[1]University of Manitoba,
Winnipeg, Manitoba, Canada
{khalad, irani}@cs.umanitoba.ca

[2]Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
david.ahlstroem@aau.at

ABSTRACT

Exploring information content on mobile devices can be tedious and time consuming. We present Around-Device Binning, or AD-Binning, a novel mobile user interface that allows users to off-load mobile content in the space around the device. We informed our implementation of AD-Binning by exploring various design factors, such as the minimum around-device target size, suitable item selection methods, and techniques for placing content in off-screen space. In a task requiring exploration, we find that AD-Binning improves browsing efficiency by avoiding the minute selection and flicking mechanisms needed for on-screen interaction. We conclude with design guidelines for off screen content storage and browsing.

Author Keywords

Around-device Interaction; Off-screen Discretization; Data Analytics; Visual Analytics.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Interaction styles.

INTRODUCTION

Mobile devices are popular portals for interacting with large information repositories. Estimates suggest that over 40% of smartphone owners browse and research products on their mobiles before making a purchase, 60% of last minute hotel bookings are made on portable devices, and search on mobiles will generate 27.8 billion more queries than on desktops by 2016 [15]. Tasks with high information bandwidth often require that users quickly browse and compare the breadth of available choices before making a decision. On mobile interfaces such tasks require many minute operations, such as flicking through screens, and opening and closing items of interest, resulting in less efficient information exploration and browsing.

Ideally, mobile devices could be augmented with a virtual interactive plane that extends beyond their physical form factor for storing, browsing and retrieving data. Instead of flipping through screens of icons or through lists of items the user can bin frequently used or interesting items in off-screen space for fast access. This may be possible with the next generation of mobiles that can sense user input around the device, in thin air [1, 2]. Shifting input intensity away from the screen, into a much larger space around the device, provides larger proxy objects for interacting with smaller on-screen items, minimizes the likelihood of having the input hand occlude important screen content, and reduces the need for frequent repetitions of small on-screen manipulations. We propose and study this idea by introducing Around-Device Binning, or *AD-Binning* (Figure 1).

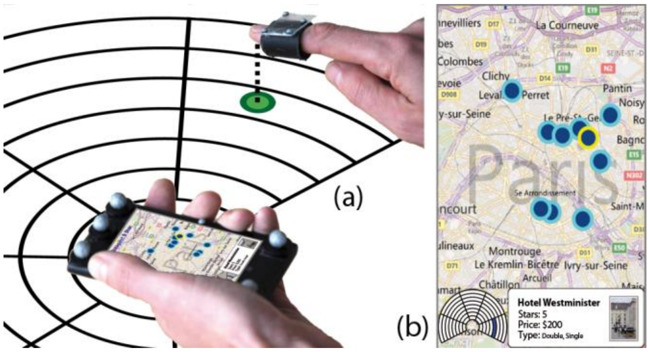


Figure 1. To make a hotel reservation, a user's query puts hotel information in AD-Bins around the mobile device. As (a) the finger hovers on an AD-Bin (b) its content is shown on the screen allowing the user to browse and compare alternatives.

While the space around a mobile device can be leveraged for different purposes [2, 9, 11], our contribution is in the design of an interactive method, AD-Binning, for directly storing and retrieving virtual content around the device. In a task requiring content browsing for making a decision, participants were more efficient with AD-Binning than with on-screen exploration. This improvement was a result of a systematic design process addressing questions such as:
º what are suitable methods for placing items off-screen?
º what selection methods provide efficient retrieval? and
º how small can targets be before affecting performance?

Our contributions include: 1) AD-Binning, a novel technique for storing and browsing content through direct interaction with around-device space; 2) an evaluation of methods for placing content off-screen; 3) the design and study of suitable around-device selection techniques; 4) methods for discretizating space around the device into bins; and 5) a demonstration of AD-Binning with an information exploration and decision-making task.

AD-BINNING DESIGN FRAMEWORK

AD-Binning is inspired by earlier work demonstrating that around-device input is valuable for interacting with small form-factor devices [7] and for extending the input vocabulary of mobiles [10, 11]. Device manufacturers are considering adopting around-device sensing methods in the next generation mobile devices [1]. Unlike most prior work on around-device interaction [2, 7, 9, 10, 11, 12], we focus on direct interaction with off-screen content. Such an interaction style presumes that the mobile's interaction plane extends beyond the physical boundaries of the device [4] and users can directly point to retrieve items. Prior work on a class of interactions involving around-body input [3, 13] suggests that users can leverage their spatial abilities to efficiently recall items through mid-air pointing. We expect similar benefits for AD-Binning. We examine additional prior work to frame AD-Binning's design factors.

Design Factors

Several key factors influence the design of AD-Binning. We explore these factors in relation to prior work.

Selection methods: AD-Binning allows users to explore content by letting the user move their finger in the space around the device. This facilitates rapid item browsing. However, a selection is required to retrieve an item and put it into focus for more details. Researchers have designed similar mechanisms for triggering a selection when iterating through items using auxiliary input channels such as pressure and tilt. These include dwelling on an item [8], quickly releasing a button [17] or lifting the finger [17]. AD-Binning facilitates item selection through two general methods: interaction on the device for triggering selection (touch or back-tap) or micro-gestures in mid-air around the device. We investigate the suitability of both these methods.

Bin size: AD-Binning relies on direct off-screen pointing to place and retrieve items, a task influenced by Fitts' law [14]. However, AD-Binning provides the advantage that items around the device can take on large sizes to compensate for the small size commonly seen on mobile devices. However, little is known of how small targets can be without affecting performance with this technique. We investigate suitable bin sizes to facilitate accurate selection.

Visual feedback: To get rapid and accurate access to around-device items, effective on-screen cues are need to point at (a) off-screen items [4] and (b) the user's moving finger. Overviews of the entire workspace have shown slightly better performance for direct off-screen pointing [4] than visual cues such as Wedge [5]. The differences between these visual techniques are affected by regions in which off-screen items are placed (performance with Wedge is non-uniform across the viewport). Our design of AD-Binning uses an overview to show relative item positions and the user's finger in AD-Space.

Space discretization for bins: Closely tied to the input range and target size is the method for breaking up the around-device space into bins, or space discretization. With auxiliary input streams (pressure and tilt) input discretization leads to better control [8, 17, 19]. Due to the bio-mechanical limits of the arm and difference in control at extreme arm ranges we examine the effect of applying different discretization methods to the task of placing and retrieving around-device items.

Input range around the device: The bio-mechanical properties of the human arm dictate that on average users can extend their arm to about 60cm [16], limiting how many items can be placed off-screen. Little is known about this range when the arm moves around the device, i.e. the right arm on the left side will have a smaller range. We capture this range in a pilot prior to our studies.

Ideal binning locations: Prior work has shown that pointing at items placed in corners around the device is less effective and accurate than pointing at items to the sides [4]. These results were not obtained by evaluating the entire range for placing items and therefore more knowledge of ideal locations can assist in the design of AD-Binning.

Binning methods: Spatial memory and proprioceptive feedback can assist in retrieving information that is laid out spatially [3, 18, 20]. Ideally, information can be placed, or binned, using techniques that leverage this capability.

Mode switching: AD-Binning requires mode switching to differentiate around-device input from other accidental gestures in space. Mode switching could be explicit, wherein the user sets the device in bin-mode as needed. Alternatively, advanced sensing mechanisms could distinguish users' fingers in space separately from other items around the device. We do not examine mode switching mechanisms in this paper.

AD-BINNING APPARATUS AND INPUT RANGE

In our work, we assume that finger tracking in 3D around the device will become possible [1]. We emulate such a system using a Vicon MX system with eight cameras (T-Series) to track participants' hand movements (Figure 2a). We placed markers on a smartphone (Nokia Lumina 800, size = 48.3×80.59mm, resolution = 480×800 pixels) and on a Velcro loop worn on the right-hand index finger (Figure 1a). A Windows Presentation Foundation server application transferred tracking data every 10ms from the Vicon over Wi-Fi to the experimental software (Silverlight Windows Phone application) running on the smartphone. Advised by previous work [3] on how performance (task speed and accuracy) drastically suffers as the input space extends from 2D to 3D, our current implementation of AD-Binning only considers the space defined by the plane around the device: all interactions above or below the plane (in the z-direction) are projected on the interaction plane (Figure 1a). Future work will investigate the use of 3D space to layer items.

To determine AD-Binning's input range we asked two female and four male adults (these and all subsequent participants were right-handed and stood in our experiments) to hold a smartphone in their left hand and to 'draw' a half-circle around the device with their right hand, going from the left to right, and then back again five times. Participants were asked to perform without reaching their maximum distance. The collected movement data resulted in three design decisions: 1) 40cm was comfortably within reach for all participants and we used this value as the maximum input range; 2) points within ¾ of a full circle are within reach, and we use this to map bins into a circular layout; 3) we split the circular space into five sectors mapped to cardinal directions (North, North-East or labeled as Top, Top-Right) to facilitate spatial recall of items (Figure 2b). More than five sectors results in to small items in the inner circle, leading to inefficient selection in these regions. Our exploration was based on these design choices and does not limit the use of other parameter values based on user preference and arm-length.

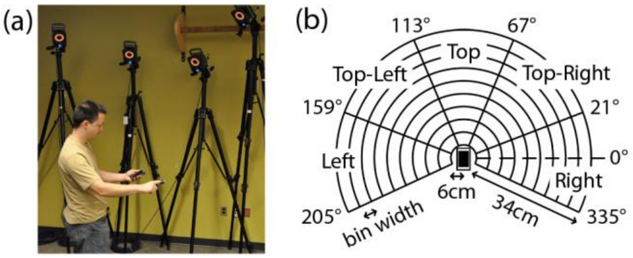


Figure 2. (a) Vicon apparatus. (b) The AD-Binning space.

EXPERIMENT 1

In our first experiment we identified suitable selection techniques, the minimum bin size for efficient item selection, and appropriate around-device space divisions methods. We split the experiment in sessions A and B to reduce experiment length and the complexity of the analysis, and to focus on a few design parameters at a time.

Selection Methods

Little is known about the specific methods for selecting items in around-device space. We grouped our selection techniques into methods that take place on or off the device. We settled on six candidate methods – two performed on the device with the non-dominant hand, and four using the dominant hand and its pointing finger in the air.

º *Tap* does not restrict the on-screen tapping area (as would be necessary when interactive elements are presented on the screen).

º *BackTap* is based on reading the device's accelerometer data and tapping the back of the device using the index finger of the device hand. After experimenting with various thresholds we found 0.15g (gravitational units) to be suitable for BackTap detection. The obvious advantage of BackTap over Tap is that it eliminates the risk of invoking interactive items on the screen during a selection. Conversely, BackTap cannot be used when the device is placed on a table.

º *Dwell* is often suggested in the literature as an alternative to click, e.g., in eye-gaze input. Our dwell time was 600ms.

º *LiftOff* requires an active movement raising the pointing finger. A change in z-position >30mm between two consecutive time cycles triggers a LiftOff. An alternative to LiftOff is to push down, which we did not test as both behave similarly.

º *Pierce* assumes an imaginary horizontal interaction plane defined by the mobile device, which the finger needs to 'pierce' to make a selection.

º *DownUp* uses a down-up motion (30mm down, and up) inside a bin to trigger a selection. The two-stage motion, up and down, allows for a backoff possibility to cancel a started selection, similar to clicking an on-screen button with a mouse. This is the only method with a possibility to reverse in mid-course of the selection.

Session A – Selection Methods and Bin Size

Participants, Task and Experimental Design

Twelve daily computer and touch screen users (3 female) aged 20 to 39 years participated. With short breaks and practice trials, each session lasted around 45 minutes.

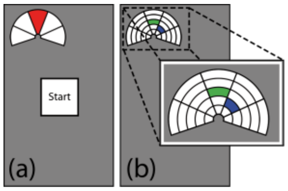


Figure 3. (a) Trial start screen. (b) Visual feedback overview.

A start button and a small overview (1.61cm wide) are displayed on the screen (Figure 3a). A red marker in the overview highlights one of the five sectors to indicate the direction to the next target bin (we chose to indicate direction before trial start to minimize visual search and unaimed arm movements at trial onset). The participant presses the start button with the right-hand index finger to begin the trial. (Participants were not allowed to time-optimize by first moving their tracked index finger to the estimated target location and then starting the trial with their left thumb. They were also restricted to use their dominant hand for interaction as allowing both hands could add confounds to the results.) The overview shows the target in green, and a blue cursor in the overview follows the tracked finger (Figure 3b). A correctly performed selection action ends the trial and loads the start screen for the next trial.

Session A used a 6×4 within-subjects design for factors *selection method* (Tap, BackTap, Dwell, LiftOff, Pierce, DownUp) and bin *widths* (68, 38, 26, 20mm). The four bin widths were obtained by dividing each sector into 5, 9, 13 or 17 equally wide bins. Participants performed ten repetitions of each *selection method-width* combination, resulting in a total of 240 trials per participant. Participants completed 20 random practice trials before the test trials.

The presentation order of the six *selection methods* was balanced among participants using an incomplete Latin

Square. The order of bin *widths* was randomized for each *selection method*. We kept the distance to the target bin constant (230mm) by only using the middle bin in each bin sector. Two trials from each *method-width* combination were located in each of the five bin sectors (Left, Top-left, Top, Top-right and Right – Figure 2b). The order of target location was randomized. No feedback was given when participants selected a non-target bin or if the intended selection action was not detected. Trials were only terminated after a correct selection occurred in the correct target bin. We asked participants to perform each trial as quickly and accurately as possible. After completing all methods participants rated them based on preference.

Results

We used a repeated measures ANOVA and post-hoc pairwise comparisons to analyze trial times. We used Friedman tests with Wilcoxon tests for post-hoc pairwise comparisons to analyze error rates (number of trials with incorrect selections divided by the total number of completed trials). Post-hoc pairwise comparisons were Bonferroni adjusted (α-level = 0.05). The same tests were used in all experiments unless otherwise noted.

Error rate: The overall error rate was 11.7% (382 of 3262 trials contained one or more undesirable selections before the target bin was selected). Figure 4a shows the mean error rates. *Selection method* had an effect on error rate ($\chi^2(5,N=12) = 30.44$, $p < 0.0001$) and pairwise comparisons showed that DownUp caused significantly fewer errors than BackTap, LiftOff and Pierce. There were no other significant differences. DownUp's low error rate is due to its twofold accuracy requirement: 3cm down and 3cm up in the same bin without veering into an adjacent one.

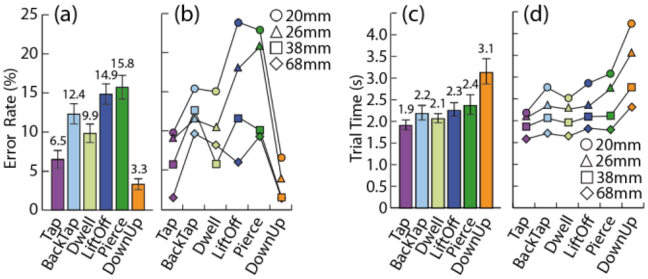


Figure 4. (a) and (b) Mean error rates. Error bars: ± 1 S.E. (c) and (d) Geometric mean trial times. Error bars: 95% CI.

We found error rates of 6.7, 8.7, 13.5, and 17.1% for 68, 38, 26, and 20mm bins, respectively, and there was also a significant effect of *width* on error rate ($\chi^2(3,N=12) = 28.30$, $p < 0.0001$). Pairwise comparisons showed that 68mm bins caused fewer errors than both 26 and 20mm bins and that 38mm bins caused lower error rates than 20mm bins. There was no statistically significant difference between the two largest and between the two smallest bin sizes. When comparing how the *selection methods* performed at each *width* (Figure 4b) we found a significant difference at each *width*, but only significant post-hoc pairwise comparisons at the smallest width, where DownUp caused fewer errors than Pierce and LiftOff.

Trial time: Trial times were positively skewed and we performed a logarithmic transformation (which resulted in distributions close to normal) before analyzing the data.

Selection method and *width* had significant effects on trial time ($F_{5,55} = 28.6$, $p < 0.0001$, $\eta^2 = 0.72$ resp. $F_{3,33} = 210.9$, $p < 0.0001$, $\eta^2 = 0.95$). Across selection methods, the geometric mean trial times (i.e., the antilog of the mean of the log-transformed data) ranged from 1.9s for the largest 68mm bins to 3.1s for the smallest 20mm bins. As a result of the increased accuracy demand, trial times increased by about 15% for each decrement in bin width. Post-hoc pairwise comparisons showed that all bin sizes differed.

Figure 4c shows the geometric means for each *selection method*. Post-hoc pairwise comparisons showed that DownUp was slower than all other selection methods and that Tap was faster than Pierce and LiftOff. There were no other statistically significant differences between the methods. The significant *method*×*width* interaction ($F_{15,165} = 2.8$, $p < 0.01$, $\eta^2 = 0.16$) plotted in Figure 4d provides deeper insights. Except for DownUp, all methods performed about equally well at 68 and 38mm bins. With 26mm bins though, we see marked peaks for Pierce and DownUp, and moderate, similar increases in the other methods. Only with the smallest bins do BackTap and LiftOff lose ground against Tap and Dwell.

Preference ratings: According to overall preference, 9 of 12 participants rated Tap to be the best, two preferred the LiftOff method and one favored BackTap.

Summary – Session A

Our results indicate that performance – in particular errors – degrades significantly after the 38mm bin size (9 bins). We suggest that for AD-Binning, targets should not be any smaller than this size. While Tap and Dwell appear to have the least errors and a trend toward faster selection times, these may not be practical in all applications. For example, dwelling may conflict with object browsing, and Tap should only be restricted to a specific on-screen target. We continue our exploration with BackTap and LiftOff as our on-device and off-device selection methods. The same participants were recruited for Session B, providing a certain level of expertise with AD-Binning.

Session B – Space Discretization & Binning Locations

As indicated above, prior studies [4, 6] have suggested an accuracy trade-off in mid-air pointing with targets distant from a reference, in our case the edge of the device. This led to our evaluation of different around-device space division or discretization methods.

The Uniform discretization (Figure 5a) divides the available space into nine equally sized bins of 37.78mm. In the

Distance Dependent discretization (Figure 5b) the inner bin is 27.2mm wide and the following bins are allotted an additional multiple of 2.64mm according to their position from the inner bin. Thus, the outer bin, which is located eight positions away, is 27.2+8×2.64=48.32mm wide. We also included a fisheye discretization technique (Figure 5c) that uses a hysteresis function similar to [19, 21] to dynamically add and remove extra space on both sides of each bin (except the first and last bins). The active bin expands to 75.28mm, its two neighbors expand to 50.28mm, and the remaining bins are 27.36mm wide. We controlled distance by dividing the available radial distance of 340mm in equal distance ranges, D1, D2, D3 and D4 (Figure 5d). A random number within the desired range was drawn and the bin at this distance was set as the next target, belonging to the corresponding distance range.

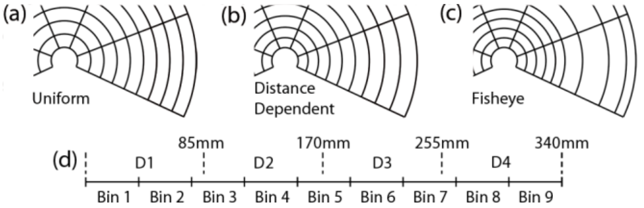


Figure 5. Discretization techniques and distance mapping.

Task and Experimental Design

All task procedures were the same as in Session A.

Session B used a 3×2×5×4 within-subjects design for the factors *discretization* (Uniform, Distance dependent, Fisheye), *selection method* (BackTap, LiftOff), *sector* (Left, Top-Left, Top, Top-Right, Right), and *distance* (D1, D2, D3, D4). Participants performed 360 trials: three repetitions for each combination of factor levels. We counterbalanced on discretization technique and half of the participants started with BackTap first. Participants completed 20 random practice trials and then 40 timed trials with each combination of discretization and selection methods.

Results

Error rate: BackTap had a significantly lower mean error rate than LiftOff (7.4% vs. 10.2%, Wilcoxon test: $Z = -2.7$, $p < 0.01$). We also found significant effects for *discretization* ($\chi^2(2,N=12) = 7.2$, $p < 0.05$) and *distance* ($\chi^2(3,N=12) = 15.3$, $p < 0.01$), but not for *sector*. Pairwise comparisons showed that the Fisheye, with a mean error rate of 6.6%, caused significantly fewer errors than both the Uniform and Distance dependent discretizations (error rates: 9.6% and 10.2%, respectively). Pairwise comparisons between distances showed that bins in distance range D1 caused more errors than bins in D2. There were no other pairwise statistically significant differences.

Interestingly, the Fisheye discretization reveals an overall equalizing effect over all distances. In the other two discretizations (Figure 6a) performance in D4 and D1 degraded, possibly due to reduced motor accuracy and smaller arc lengths, in the far and close bins, respectively. The Fisheye discretization also evened out the error rates between selection methods. The overall higher error rate with LiftOff is a result of poor performance when combined with Uniform and Distance Dependent discretization (Figure 6b). When extra space is added to the 'current' bin, as in the Fisheye, LiftOff performs as well as BackTap.

Trial time: As in Session A, trial times were positively skewed and we applied a logarithmic transform (with distributions close to normal) before analyzing the data. The geometric mean trial time was 2.19s for BackTap and 2.20s for LiftOff. Across the two *selection methods*, the geometric means for the three discretization techniques were 2.24, 2.17 and 2.17s for the Uniform, Distance dependent and Fisheye, respectively. We did not find main effects for *selection method* or *discretization*, but there was a main effect for *sector* ($F_{4,44} = 8.0$, $p < 0.0001$, $\eta^2 = 0.42$) and for *distance* ($F_{3,33} = 33.6$, $p < 0.0001$, $\eta^2 = 0.75$).

Post-hoc pairwise comparisons between distances showed that bins in D4, with a geometric mean trial time of 2.49s, were significantly slower to select than bins located elsewhere. Bins in D2 were the fastest (1.97s). There were no statistically significant differences between bins in D1 or D3 (2.18 vs. 2.16s).

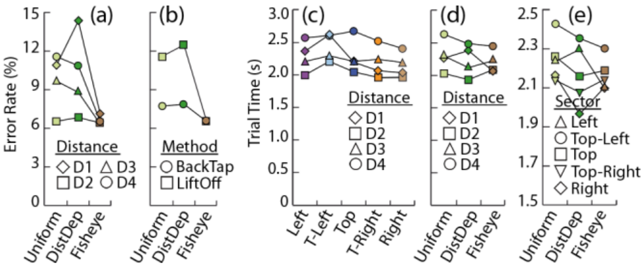


Figure 6. (a) and (b) Interaction effects, error rate. (c), (d), and (e) Interaction effects, trial time.

Post-hoc pairwise comparisons between sectors showed that the Right and Top-Right sectors, with geometric means of 2.08s and 2.12s, respectively, were faster than the Top-Left sector which was the slowest at 2.36s. There were no other statistically significant differences between any other sectors. A significant *sector*×*distance* interaction ($F_{12,132} = 3.9$, $p < 0.0001$, $\eta^2 = 0.26$) (Figure 6c), identifies D1 and D2 having marked peaks as the main sources for the overall poor performance in the Top-Left sector. We attribute these problems to occlusion: presumably, keeping the wrist at a natural angle when targeting Top-Left bins close to the device causes the hand to occlude parts of the screen and the visual feedback provided by the overview.

We also observed a *discretization*×*distance* effect ($F_{6,66} = 7.4$, $p < 0.0001$, $\eta^2 = 0.40$, Figure 6d). As with errors, the Fisheye had an equalizing effect on trial time. It is notable that trial times for bins close to the device (D1) drop as a result of the Fisheye expansion. Comparing Distance Dependent to Uniform discretization reveals a clear negative effect of removing space from D1-bins (confirming Session A's result that bins should be ≥ 38mm).

The significant *discretization×sector* effect ($F_{8,88} = 2.7$, $p < 0.05$, $\eta^2 = 0.19$, Figure 6e), reveals that the Fisheye also equalized performance between sectors. It reduced selection times in the slow Top-Left sector, but also in the Left sector. It is also notable that the Distance dependent discretization improved performance in the Right sector.

Preference ratings: Eleven participants rated Fisheye as the preferred technique and one rated Distance Dependent as the best. Seven rated the Uniform discretization as their least preferred technique.

Summary – Session B

We observe that the Fisheye discretization had an overall equalizing effect on error rates and trial times, across selection method, distance and sector. Our following experiments use the Fisheye for dividing the around-device space. Unexpectedly, selecting targets in the closest distances was less accurate and less efficient. Due to constrained movements with crossing arms, areas left and top-left of the device are generally more cumbersome.

EXPERIMENT 2 – BINNING & RETRIEVAL

Binning items could conceivably be done at any time. The user could quickly place an application icon, contact entry or web-bookmark in a system wide bin-collection for long-term storage and fast access. Binning could also be application dependent and serve more short-term purposes, such as browsing the results from a query or to manage a sub-set of items of temporary interest (e.g., yesterday's emails). In this experiment we compare binning techniques that provide varying degrees of user-control:

º *Automatic* provides no user control on item placement. The system assigns each item to an empty bin. Assignment can be random or based on item properties (e.g., a name, time stamp, color). Items are binned in a batch, either initiated by the user (e.g., by shaking the device) or automatically triggered through a query interface.
º *Tap-and-Bin* allows full user control. The user picks items, one by one, tapping their on-screen representations, and then, guided by the cursor in the on-screen overview, moves the hand to the desired AD-bin to 'drop' it using a LiftOff gesture. Tap-and-Bin may be time consuming with many items but facilitates individual placement strategies for improved recall. The direct acquaintance with each item in combination with the following arm movement may also help develop valuable proprioceptive memory linkages.
º *Flick-and-Bin* provides semi-automatic binning that uses the flick direction to choose a sector and then places the flicked item in an empty bin that is closest to the device in that sector. If all bins are filled in the directed sector the user has to flick in another direction. The on-screen overview provides dynamic sector highlighting during the flick. The automatic 'first-empty' strategy makes Flick-and-Bin fast at the expense of user control. A more elaborate version could map flick-distance to bin-distance for full user control.

Participants, Task and Experimental Design

Twelve daily computer and touch screen users (3 female) aged 20 to 39 years participated. Five had participated in Experiment 1. Participation lasted approximately 30 minutes (including short breaks and practice).

Phase 1 of a trial consists of binning multiple icons (6 or 12); Phase 2 involves retrieving three of them. With Tap-and-Bin and with Flick-and-Bin the participant taps a start button, the next icon to bin in the trial is displayed on the screen, and timing starts. With Tap-and-Bin the participant taps the icon, moves the hand into AD-space, then bins the icon in an empty bin using a LiftOff, and timing ends. With Flick-and-Bin, the participant flicks the icon towards a sector with an empty bin and timing ends. The on-screen overview, where empty bins are yellow and occupied are blue, provides dynamic feedback throughout the binning activity. Flick-and-Bin forces items to be binned in the directed sector with inner bins filled first. Tap-and-Bin provides the most flexibility in terms of item placement. No flexibility is provided with Automatic binning where the system does the binning (i.e., Phase 1 takes no time) and automatically fills inner bins (starting from the right sector going left) before more distant bins are used.

A dialog box announces Phase 2 when all icons are binned. Dismissing the dialog box displays three random icons from Phase 1 for 10 seconds as a preparation for the upcoming three retrievals. Showing items prior to retrieval is representative of a real task where users know ahead of time what items they are looking for (such as during a search task). When the three icons disappear, timing begins, and the participant starts the first retrieval. As the retrieving finger moves beyond the screen border, the overview indicates its current location with a red marker. The bin content is shown next to the overview. When the correct bin has been found, the retrieval (and timing) ends with a LiftOff in the corresponding bin. After retrieving all three target icons, the binning Phase of the next trial starts.

The experiment used a 3×2×3 within-subjects design for factors *technique* (Automatic, Flick-and-Bin, Tap-and-Bin), *set size* (6 or 12 icons to bin), and *retrieval* (first, second, third in each trial). With three retrievals per trial, three trial repetitions with each technique and set size combination each, participant performed a total of 54 retrievals. The order of *technique* was counterbalanced between participants and set sizes were presented in a random order for each technique. We used the Fisheye discretization and five sectors with three bins each. Participants had two practice trials with each technique. Icons were randomly chosen from a set of 180 similarly styled images. No icon appeared in two consecutive trials.

Results

Binning Phase

Binning strategies: after the experiment, all participants indicated that they tried to bin items strategically. With

Tap-and-Bin and Flick-and-Bin, most participants categorized items (e.g., 'eatables', 'computer stuff', 'red ones', etc.) in sectors (participants did not know in advance what items to bin). As expected, it was easier to apply this strategy with six items than with twelve. Participants placed items in inner bins before the outer ones. Trials with six items provided more flexibility regarding bin choice but participants clearly avoided using the Left and Top-Left sectors (Figure 7a and b).

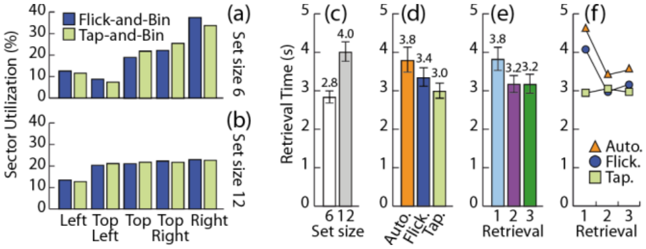


Figure 7. (a) and (b) Sector utilization. (c), (d), and (e) Geometric mean retrieval times. Error bars: 95% CI. (f) Technique×retrieval interaction.

Binning time: As expected, Automatic binning took no placement time. Participants spent on average 2.9s to bin an item with Flick-and-Bin and slightly longer, 3.3s, when using Tap-and-Bin. With a mean trial time of 2.2s for LiftOff-selections in Experiment 1 (i.e., the same movement and gesture required by Tap-and-Bin), we see a strategizing overhead of 1.1s for Tap-and-Bin. Allegedly participants used the same strategies for both techniques but the flick gesture in Flick-and-Bin took on average 0.4s.

Retrieval Phase

Error rates: In 42 of the 648 collected trials (6.5%) participants made at least one, and at most four, erroneous selections before the prompted item was selected. With all three placement techniques using LiftOff as the selection method, we found no statistically significant difference in error rates between techniques. There was also no statistically significant difference in error rates between *set sizes* or *retrievals*.

Retrieval time: Retrieval times were positively skewed and we applied a logarithmic transform (with distributions close to normal) before analyzing the data.

The geometric mean retrieval times for each *set size*, *technique* and *retrieval* are shown in Figure 7c, d and e. We found significant main effects for all factors (*set size*: $F_{1,11}$ = 100.2, $p < 0.0001$, $\eta^2 = 0.90$; *technique*: $F_{2,22} = 7.3$, $p < 0.01$, $\eta^2 = 0.40$; *retrieval*: $F_{2,22} = 8.3$, $p < 0.01$, $\eta^2 = 0.43$). As expected, the larger *set size* required more searching than the small set size, and thus took longer time. Post-hocs between *techniques* showed that Tap-and-Bin was significantly faster than Automatic and that there were no other pairwise statistically significant differences. Post-hocs between *retrievals* showed that the first retrieval was significantly slower than the other two. The second and third retrievals did not differ.

The significant *technique*×*retrieval* interaction ($F_{4,44} = 5.9$, $p < 0.001$, $\eta^2 = 0.34$, Figure 7f) reveals that the overall advantage of Tap-and-Bin (21% vs. Automatic, 12% vs. Flick-and-Bin) is mainly a result of exceptionally fast first retrievals. With previous placement analysis showing that participants did not make use of the possibility to leave the inner bins empty with Tap-and-Bin, it is particularly interesting to note the large difference between Tap-and-Bin and Flick-and-Bin in the first retrieval. With no inner bins empty, the only difference between the two techniques is the amount of physical activity required to do the binning, a short flick for Flick-and-Bin, moving the arm and a LiftOff gesture for Tap-and-Bin. Apparently, the greater physical activity needed for Tap-and-Bin fostered spatial memory. In the first retrieval with Flick-and-Bin and Automatic, participants had to rely more on the visual overview and search. The position information participants gained during this first search was then utilized in later retrievals to improve performance to Tap-and-Bin's level.

Summary

Our results suggest that the overhead involved in manually binning items as in Tap-and-Bin is compensated by improved retrieval times due to enhanced spatial encoding. Spatial enforcement of item locations is also present while searching for items in AD-Bins: retrieval performance improved after having selected the first item in Automatic and Flick-and-Bin, as participants mentally recorded positions of subsequent items to retrieve.

Results from the above studies suggest that AD-Binning can facilitate selection of reasonably large items (Exp. 1A), where errors and selection times across distance and sectors can be equalized using a space discretization technique such as the Fisheye (Exp. 1B). Furthermore, exploring AD-Binning space enhances spatial encoding of item positions around the device (Exp. 2). These results inform the selection of suitable design parameters for an efficient AD-Binning technique.

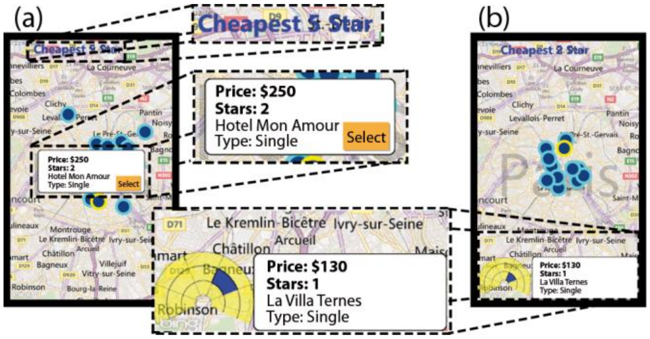
EXPERIMENT 3 – ANALYTIC TASK

With knowledge of how the various design parameters influence performance from the previous experiments, we next demonstrate and evaluate a practical AD-Binning usage scenario for browsing information content.

Participants, Task and Experimental Design

Twelve daily computer and touch screen users (3 female) aged 18 to 35 years participated. Two were new and had not participated in any previous experiment. Participation lasted approximately 45 minutes (including breaks and practice). The task simulates a frequent situation where the user has queried a system for information. In our case, a geographic tourist portal for hotel reservations where the query results are displayed on a map. Issuing the query can result in items being placed automatically in around-device space, and ready for retrieval.

A trial starts with the screen displaying a prompt to search for the cheapest n-star hotel (n=number from 1 to 5). After reading the text, the participant taps a start button and trial time starts. The next screen shows a city map with a set of circular markers (ø 5mm) representing various hotels. We place the search criteria at the top of the screen as a constant recourse (Figure 8a). In the 'on-screen' condition, the price and rating (number of stars) for a hotel are displayed in a callout box that opens when the marker is tapped (Figure 8a). Participants were informed that they could either close the box with a tap on the map or fill the box with new information by directly tapping on another marker. When the participant believes having found the hotel satisfying the search criteria, the trial ends with a tap on the button (1×0.7cm) in the callout box. If correct, the trial time stops and the text prompt for the next trial is displayed. If incorrect, an error message pops up which blocks further input for one second before it automatically fades away. After that, the search for the correct hotel can continue. Panning and zooming are fully enabled.


Cheapest 5 Star

Price: $250
Stars: 2
Hotel Mon Amour
Type: Single
Select

Cheapest 2 Star

Price: $130
Stars: 1
La Villa Ternes
Type: Single

Figure 8. (a) On-screen interface with 10 hotels at low density. (b) AD-Bins with 15 hotels at high density.

With AD-Binning, we use the automatic binning method such that 'proxies' to hotel markers are placed in random AD-bins. The AD-space is divided into five sectors, with a total of 5, 10 or 15 bins depending on the condition (see below), and uses the Fisheye discretization. The participant browses hotels in off-screen space by moving the index finger between bins. At bin-entry, the corresponding hotel marker is highlighted and the hotel information is shown next to the bin-overview at the bottom of the screen (Figure 8b). To select a hotel, the user performs a LiftOff inside the desired bin. The trial prompt, timing and error notifications work as previously described. On-screen panning and zooming are fully enabled. As this task primarily involves browsing, we opted for the automatic binning method over manual binning which is better suited for more complex tasks that involve setting aside objects for later retrieval.

The experiment used a 2×2×3 within-subjects design for factors *interface* (on-screen, AD-Binning), marker *density* (low, high), and *number of items* (5, 10, 15). Participants performed five repetitions for each combination of factor levels, for a total of 60 timed trials per participant (five practice trials were given per interface). Combinations of density and number of items were presented in random order within each interface. Six participants started with AD-Bins, six with on-screen browsing. In low-density conditions all hotel markers were positioned at random positions within 1.974cm of the map/screen center. The high-density conditions used 0.987cm. At least two hotels with the requested number of stars existed in each trial. Hotels, prices, stars, and marker positions were otherwise completely randomized.

Results

Error Rate

In 90 of the 720 collected trials (12.5%) participants made at least one, and at most five, erroneous selection before finding and selecting the correct hotel (44 trials with on-screen, 46 with AD-Bins). Neither *interface* nor *density* influenced the error rate (Wilcoxon tests), but *number of items* did ($\chi^2(2,N=12) = 10.8$, $p < 0.01$). Post-hocs showed differences between 15 and 5 items (46 vs. 19 trials) and between 15 and 10 items (46 vs. 25 trials). Naturally, with more items to manage and to compare, the risk of making a mistake increases.

Trial Time

AD-Binning, with a mean trial time of 17.5s (s.d. 9.3), was significantly faster ($F_{1,11} = 43.9$, $p < 0.00001$, $\eta^2 = 0.80$) than the on-screen interface with a mean trial time of 27.6s (s.d. 17.2). There was no statistically significant difference between high and low *density* but *number of items* significantly influenced trial time (Greenhouse-Geisser corrected, $F_{1.12,12.31} = 164.6$, $p < 0.00001$, $\eta^2 = 0.94$) with post-hoc comparisons showing differences between all factor levels.

Number of items interacted with both *interface* ($F_{2,22} = 24.5$, $p < 0.00001$, $\eta^2 = 0.69$) and *density* ($F_{2,22} = 3.7$, $p < 0.05$, $\eta^2 = 0.25$) (Figure 9a and b). Time savings with AD-Binning increased disproportionally with the number of items: 27.9% with five items, 31.9% with ten, and a whopping 40.2% with 15 items. Overall, AD-Binning was 35.8% faster. Across *interface*, there were no statistically significant differences between low and high *density* in conditions with 5 and 10 items but performance deteriorated with 15 items at high density (Figure 9b). Presumably, this was mostly caused by the increased need for elaborate pan/zooming and closing of the callout actions in the on-screen condition (however, we note that there was no significant 3-way interaction).

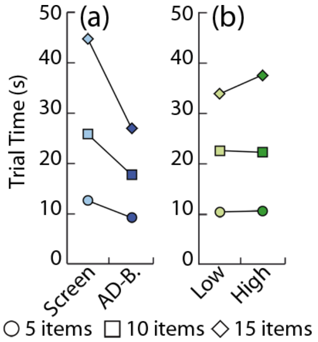


Figure 9. (a) Numbers of items×interface. (b) Numbers of items×density.

Preference Ratings

Three of twelve participants preferred the on-screen interface, eight preferred the AD-Binning and one was undecided. The most frequent reason for preferring AD-Binning was how it helped recalling the rough location (sector/'quadrant') of the 'current best answer'.

Summary

In comparison to on-screen input, AD-Binning reduces information browsing time for three reasons. First, AD-Binning is in 'browse' mode by default. Retrieving object information involves hovering or sliding the finger between bins. In contrast, on-screen consumes at least two steps: i) tap on an icon to pull-up information, and ii) tap again to close the callout box or to retrieve information from another marker. Second, AD-Binning target sizes can be significantly larger than those on-screen. In our example, queries with 5 items used five bins, each with a larger space then when the query had 15 items. Finally, participants exploited spatial abilities with AD-Binning. They would cache in memory the best bin location satisfying the query criteria and update in memory this bin location only when the next best item was available. While this happened with automatic placement, in a full manual placement reliance on spatial memory would be even stronger as indicated from results of Experiment 2.

DISCUSSION

We summarize our main findings and present them as around-device binning guidelines, discuss other applications that can benefit from our guidelines and conclude with some limitations of our investigation.

Design Considerations

Our results offer the following guidelines to designers for interfaces similar to AD-Binning:

º *Input range:* An interaction space extending 40cm beyond each side of the device is suitable when around-device interaction is focused on a horizontal plane defined by the device. A radial division and partitioning in sectors allows for comfortable reach.

º *Target Size:* Use the largest targets possible, and targets should not be much smaller than 4cm across. With a radial bin arrangement, interactions close to the device cause a higher number of errors, as bins are smaller there.

º *Ideal interaction regions:* Prioritize interactions on the same side as the dominant pointing hand, as users intuitively avoid interaction on the non-dominant side to avoid occluding on-screen visual guidance.

º *Around-device space division:* Fisheye discretization can suitably divide around-device space to provide equally efficient access to all content around the device and reduces accuracy requirements.

º *Selection methods:* On-screen and off-screen methods can be equally effective for selection. Designers can choose a selection method based on task. Finger lift-off is possible in both mid-air and when the device is resting on a surface.

º *Placement methods:* Promote spatial learning through direct 'physical contact' with off-screen space. The extra time and effort required to manually place items in around-device space pays off in item browsing and retrieval tasks. Rapid binning is possible with automatic placement methods, which can be triggered through a query.

Applications

Some obvious applications for AD-Binning including photo storage (organized in sectors denoting date, event or any other semantic information), storing and retrieving items from contact lists and bookmarking items of interest when browsing a web site. We also envision longer term applications where the user can capitalize on proprioceptive memory linkages developed over time to access content across applications or regularly issued commands, similar to CommandMaps [18]. With further development, AD-Binning could also apply to the following applications:

Item retrieval based on item organization. Our implementation of AD-Binning did not consider specific ordering of items. However, many datasets have inherent structures that AD-Binning interfaces can leverage. For example, items could be sorted based on price, alphabetically or chronologically. In an email client, items can be placed in chronological order in around-device space. This can allow the user to retrieve items immediately based on their previously developed knowledge of AD-Binning item organization.

Mixing physical and virtual bins. AD-Binning could also be used in mixed physical and virtual workspace scenarios. By tracking the position of a digital pen, AD-Binning could facilitate note taking and brainstorming scenarios where ideas and sketches are made on physical notes arranged around the device (Figure 10a). Committing the final note content and position stores the note in the corresponding AD-space for later retrieval or browsing (Figure 10b).

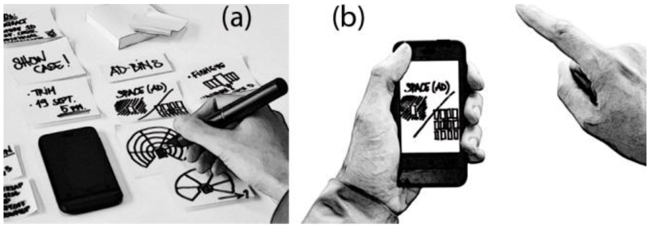


Figure 10. AD-notes. (a) Creating, arranging and storing physical notes. (b) Browsing notes stored in AD-space.

Limitations and Future Work

AD-Binning relies on a robust tracking mechanism (Vicon cameras) for around-device interaction and in 3D space. Further experimentation is needed for determining suitable design parameters for devices equipped with new sensors that track fingers in off-screen space. Our results indicate that partitioning around-device space into 45 bins (based on five sectors and the smallest target size) is a suitable strategy. However, further studies are needed to investigate other approaches such as non-circular ones or 3D-layouts that allow stacking items on top of one another. Our results are also dependent on visual feedback, which consumes space on the screen. Additional work is needed to identify whether such visual guidance can be eliminated after repeated use in a given task and application (as indicated by the results of Experiment 2). Finally, our automatic placement strategy inserts items in bins in a random manner. More robust layout mechanisms are needed to provide for an efficient organization of around-device items. For example, in the map application, items in one area could be assigned to corresponding relative regions in off-screen space. Additional future work will consider extrapolating our results to other forms of tasks in around-device space, such as selecting commands, bridging between physical items around the device and AD-Binning, and coupling around-device input with on-screen interaction. We are also interested in investigating the social acceptability of around-device interactions.

CONCLUSION

We have presented the design and evaluation of AD-Binning, a novel user interface for future small-screen mobile devices that will be able to sense finger movements in their vicinity. With AD-Binning the user can off-load screen items from the small screen into the larger off-screen space around the device. AD-Binning was mainly designed to support the user in analytic scenarios that require intensive browsing and comparisons between many alternatives, such as long query result lists or other information intensive situations where exploration is necessary before a decision is made. Such tasks can be laborious to perform using the interactions provided by small touch screens. With AD-Binning the user can efficiently store, browse and retrieve content through direct interactions in the space around the device.

REFERENCES

1. Avrahami, D., Wobbrock, J.O., and Izadi, S. Portico: tangible interaction on and around a tablet. In *Proc. UIST 2011*, ACM Press (2011), 347-356.
2. Butler, A., Izadi, S., and Hodges, S. SideSight: Multi-"touch" interaction around small devices. In *Proc. UIST 2008*, ACM Press (2008), 201-204.
3. Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., and Looser, J. Air pointing: design and evaluation of spatial target acquisition with and without visual feedback. *Int. J. Human-Computer Studies 69*, (2011), 401-414.
4. Ens, B., Ahlström, D., Cockburn, A., and Irani, P. Characterizing user performance with assisted direct off-screen pointing. In *Proc. MobileHCI 2011*, ACM Press (2011), 485-494.
5. Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. Wedge: clutter-free visualization of off-screen locations. In *Proc. CHI 2008*, ACM Press (2008), 787-796.
6. Gustafson, S., Bierwirth, D., and Baudisch, P. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proc. UIST 2010*, ACM Press (2010), 3-12.
7. Harrison, C. and Hudson, S.E. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In Proc. *UIST 2009*, ACM Press (2009), 121-124.
8. Hasan, K., Yang, X.-D., Bund, A., and Irani, P. A-Coord input: coordinating auxiliary input streams for augmenting contextual pen-based interactions. In *CHI 2012*, ACM Press (2012), 805-814.
9. Jones, B., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G. Around device interaction for multiscale navigation. In *Proc. MobileHCI 2012*, ACM Press (2012), 83-92.
10. Ketabdar, H., Yüksel, K., and Roshandel, M. MagiTact: interaction with mobile devices based on compass (magnetic) sensors. In *Proc. IUI 2011*, ACM Press (2011), 413-414.
11. Kratz, S. and Rohs, M. Hoverflow: exploring around-device interaction with IR distance sensors. In *Proc. MobileHCI 2009*, ACM Press (2009), 1-4.
12. Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G., and Nischt, M. PalmSpace: Continuous Around-Device Gestures vs. Multitouch for 3D Rotation Tasks on Mobile Devices. In *Proc. AVI 2012*, ACM Press (2012), 181-188.
13. Li, F.C.Y., Dearman, D., and Truong, K.N. Virtual Shelves: Interactions with orientation aware devices. In *Proc. UIST 2009*, ACM Press (2009), 125-128.
14. MacKenzie, I.S. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction 7*, 1 (1992), 91-139.
15. Mobile Commerce and Engagement Stats, http://www.digby.com/mobile-statistics/, last access Jan. 2013.
16. NASA-STD-3000 Man-Systems Integration Standards, Rev. B, 1995.
17. Ramos, G., Boulos, M., and Balakrishnan, R. Pressure widgets. In *Proc. CHI 2004*, ACM (2004), 487-494.
18. Scarr, J., Cockburn, A., Gutwin, C., and Bunt, A. Improving command selection with CommandMaps. In *Proc. CHI 2012*, ACM Press (2012), 257-266.
19. Shi, K., Irani, P., Gustafson, S. and Subramanian, S. PressureFish: a method to improve control of discrete pressure-based input. In *Proc. CHI 2008*, ACM Press (2008), 1295-1298.
20. Wang, Q., Hsieh, T., and Paepcke, A. Piles across space: Breaking the real-estate barrier on small-display devices. *Int. J. Hum.-Comp. Stud. 67*, 4 (2009), 349-365.
21. Zhao, S., Dragivecic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *Proc. CHI 2007*, ACM Press (2007), 1395-1404.