Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices

Pekka Parhi*
MediaTeam Oulu
Department of Electrical and Information Engineering
P.O. Box 4500, FIN-90014 University of Oulu, Finland
pekka.parhi@ee.oulu.fi

Amy K. Karlson, Benjamin B. Bederson
Human-Computer Interaction Lab
Computer Science Department
University of Maryland, College Park, MD
{akk, bederson}@cs.umd.edu

ABSTRACT

This paper describes a two-phase study conducted to determine optimal target sizes for one-handed thumb use of mobile handheld devices equipped with a touch-sensitive screen. Similar studies have provided recommendations for target sizes when using a mobile device with two hands plus a stylus, and interacting with a desktop-sized display with an index finger, but never for thumbs when holding a small device in a single hand. The first phase explored the required target size for single-target (*discrete*) pointing tasks, such as activating buttons, radio buttons or checkboxes. The second phase investigated optimal sizes for widgets used for tasks that involve a sequence of taps (*serial*), such as text entry. Since holding a device in one hand constrains thumb movement, we varied target positions to determine if performance depended on screen location. The results showed that while speed generally improved as targets grew, there were no significant differences in error rate between target sizes ≥ 9.6 mm in discrete tasks and targets ≥ 7.7 mm in serial tasks. Along with subjective ratings and the findings on hit response variability, we found that target size of 9.2 mm for discrete tasks and targets of 9.6 mm for serial tasks should be sufficiently large for one-handed thumb use on touchscreen-based handhelds without degrading performance and preference.

Categories and Subject Descriptors

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Measurement, Design, Experimentation, Human Factors.

Keywords

One-handed, mobile devices, touch screens, keypads, key size.

1. INTRODUCTION

Powerful handheld devices are rapidly paving their way as people's personal trusted devices. This trend is visible in the increasing capabilities of smartphones and PDAs, enabling these devices to be used for an ever-increasing variety of tasks. Interface designs that allow tasks to be performed one-handed can offer a substantial benefit by freeing a hand for the variety of physical and attentional demands common to mobile activities [13]. Furthermore, the prevalence of one-handed thumb-based device operation has been confirmed through the study of device users under mobile scenarios [6]. With their compact form, numeric keypad-based devices may be the design on the market that best supports the physical requirements for one handed use; however, generalized interaction with such devices is limited to keypad mapped menus and directional navigation, which has proven to be neither user-friendly nor efficient. Touch-sensitive screens, on the other hand, offer greater flexibility for software design, but their interfaces are traditionally designed for pen-based interaction requiring two hands. Even models which include an integrated miniaturized QWERTY keyboard are unwieldy for single-handed use and control due to their wide form factors and small keys.

However, some research towards one-handed thumb use of touchscreen-equipped handhelds has been conducted recently. Karlson et al. designed two interfaces to investigate interaction models for generalized single-handed use of a PDA; AppLens used thumb gestures for controlling an input cursor indirectly, while LaunchTile supported direct manipulation using thumb-sized targets [7]. Nesbat designed the MessagEase text entry system for small devices [12], which included a scalable soft keypad implementation that could be operated with a single hand.

Although LaunchTile and MessagEase both presented targets for direct thumb interaction, the studies of these designs did not focus on how large the targets should be. Because touchscreen widgets compete with other information for limited screen space, it is desirable to keep the dimensions of interaction targets as small as possible without degrading performance or user satisfaction. Previous studies have determined optimal target sizes for interaction with a stylus on a handheld as well as for index fingers on a desktop-sized display [3]. But to the best of our knowledge, none have considered one-handed thumb use of touchscreen-equipped handhelds.

Our goal is to develop analogous guidelines for interaction targets that maximize performance and preference during one-handed thumb-use of touchscreen-based devices. We have therefore designed and conducted a two-part study to investigate the interaction between target size and task performance, considering first single-target (*discrete*) and then multi-target (*serial*) tasks. We expect guidelines derived from the experimental results will help inform future research on interfaces designed to support one-handed use of small touchscreen-based devices.

MobileHCI'06, September 12–15, 2006, Helsinki, Finland.

* This work was done while the first author was visiting UMD as an intern during Fall 2005.

2. RELATED WORK

A number of studies that consider appropriate target sizes for touchscreen use have already been conducted both for PDAs [2,9,10,11,18] and desktop-sized touch-sensitive displays [3,16]. Unfortunately, recommendations from studies conducted to date are not strictly applicable to our work. Although previous PDA studies target the same platform we do, they focus on two-handed stylus input rather than single-handed thumb input. Studies that address desktop-sized displays, on the other hand, do consider finger-based interaction, but recommendations cannot be directly applied since (1) the tip of an index finger is typically smaller than that of a thumb, and (2) users of desktop displays do not have to hold the device as well as interact with it, and thus have different motor constraints than users of PDAs.

Investigations into appropriate target sizes for stationary tasks on a PDA using a stylus have drawn different conclusions about whether target size affects performance. MacKenzie and Zhang [9] found no difference in text entry rates between two QWERTY-based virtual keypads, one with 6.4 mm wide keys and the other with 10mm wide keys. While these targets are fairly large for stylus entry, Sears and Zha [18] confirmed and extended this finding for keys from 2.6-4.4 mm wide. However, in studying single-target selection tasks for targets between 2-5 mm, Mizobuchi [11] generally found speed and error rate improved with increases in key size, and though Brewster [2] was specifically interested in the interaction between target size and audio feedback on performance, he too found a significant improvement in throughput when targets increased from 2.5 to 5 mm.

While these results seem contradictory, they are both consistent with Fitts' model for motor movement [4], which defines movement time (MT) with respect to the distance to (or amplitude, A) and size of (W) the target as:

$$\mathrm{MT} = a + b\,(\mathrm{ID}) \qquad (1)$$

The constants *a* and *b* have been described as representing efficiency of the pointing device in question (here, the stylus on a PDA), while the index of difficulty, (ID), defined in [8] as $\log_2(A/W + 1)$, embodies the intuition that targets are harder to hit the farther they are, but easier to hit the larger they are. Thus the lower a target's index of difficulty, the easier (faster) it will be to hit. In the text entry studies of MacKenzie and Sears, the keypads scaled uniformly, which maintained constant IDs across changes in key sizes; since IDs were equal in each condition, it makes sense that performance rates were also the same. However the task designs of Mizobuchi and Brewster varied only target size, not distance, so IDs were not the same across conditions. Thus here, too, the results are consistent with Fitts' Law, which would have predicted the smaller targets would be more difficult, and thus slower to hit.

In an experiment carried out by Himberg et al. [5], subjects used the thumb of their primary hand for interacting with a soft keypad located at the edge touchscreen-enabled laptop PC. The laptop had phone back covers attached to the back of the display in order to make the interaction more similar to one-handed use of a handheld. However, instead of studying accuracy and performance for different sized targets, their goal was to explore the viability of soft keypad adaptation and the experiment was not specifically designed to account for speed of entry.

When desktop-sized touchscreen displays entered popular use, early studies were designed to better understand interaction with the new technology. Sears and Shneiderman showed novel selection strategies, such as delaying selection until the user removed his finger from the surface (lift-off), could offer access speed and accuracy that rivaled the mouse for targets as small as 1.7x2.2 mm [17]. Even so, selection times were fastest and error rates lowest for the largest targets tested (13.8x17.9 mm). In a later study of touchscreen-based keyboards, Sears et al. [16] investigated the interaction between key size and typing speed. Keys were sized between 5.7 mm and 22.5 mm, arranged in a QWERTY layout, and selected using any finger(s) from either hand. They found text entry rates increased with key size for both novice and experienced users, and that novices made significantly fewer errors on the largest keyboard vs. the smallest.

Recently Colle and Hiszem [3] manipulated size and spacing of targets for a touch-sensitive kiosk display, using a similar design to [2]. In their experiment the participants used their right hand index finger to interact with the display. Just as in [16], they found that between 20 mm and 25 mm offer the users the best balance among speed, accuracy and preference. Unfortunately, for handhelds with limited screen space, these target sizes would be too large, so obviously different guidelines have to be determined for thumb use on a small handheld device.

3. STUDY DESIGN

Motivated by the requirement for efficient text and numeric entry, the majority of previous investigations into optimal target sizes have preferred experimental designs modeled after data entry tasks. However, Colle and Hiszem [3] presented interesting results that while error rate decreased when targets increased from 10 mm to ≥15 mm for strings of lengths 4 and 10, the error rate remained constant for strings of length 1. This finding suggests that there is a difference between tasks that require selection of a single target (e.g., selecting a button, checkbox, or menu alternative), and those comprising a rapid sequence of selections (e.g., text or numeric entry). One possible explanation for the differences observed might be that users traded accuracy for speed when they anticipated a large number of selections, taking more care when the task involved only a single selection. This is supported by the fact that for all target sizes, users spent more time per character for strings of length 1.

For the purposes of our work, we term the single target selection tasks *discrete*, and multiple target selection tasks *serial*. Since both types of tasks are common to touchscreen interaction, we developed a two-part study to investigate optimal target sizes for each type of task: the discrete target phase consisted of tasks involving a single target selection, most similar to real-world tasks of clicking a button or selecting a menu alternative; the serial target phase presented users multiple-target tasks most similar to real-world data entry tasks such as numeric or text entry. Because of the limited extent and mobility of the thumb while grasping the device, for each phase we also took the location of the target on the screen into account, which has not been addressed in the previous studies for PDAs.

Colle and Hiszem [3] identified two metrics for evaluating tap accuracy. One approach is to vary the target size experimentally and then reason about viable target sizes according to hit rate. The second approach offers users small fixed-sized targets and instead derives a required target size from the raw hits distribution. The benefit of the second approach is that it also reveals hit bias with respect to the target location. Since our primary goal is to capture user accuracy in hitting actual interface objects, we modeled both phases of our study after the first approach of varying target sizes. However, for the benefit of understanding how screen location may affect error rate, and hence target size, we also tracked and report on actual tap locations.

4. METHOD

The study was divided into two phases. After completing an initial questionnaire to collect demographics and prior device use, the participants performed the discrete target phase followed by the serial target phase. After each phase, participants recorded subjective ratings of the interaction experience. Performance was assessed by both speed and accuracy of task completion across various target sizes and locations. The total session time, including instruction, both data collection phases and all questionnaires, was approximately 45 minutes.

4.1 Participants

Twenty participants (17 male, 3 female) were recruited via e-mail announcement and fliers posted in the Department of Computer Science at the University of Maryland, College Park, with the only restriction that participants were right-handed. The age of the participants varied between 19 and 42 years, with a mean of 25.7 years. Participants received $10 for their time. While 18 participants used keypad-based handhelds regularly, only 5 used touchscreen-based handhelds even occasionally. Participants were asked to rate how often they had used different interaction techniques with touchscreen and keypad-equipped handhelds using a 5-point scale (1 = never, 5 = always). With keypad-based handhelds, all participants strongly favored one-handed thumb use (μ=4.17) over a two-thumb technique (μ=2.56), and more rarely used two hands with index finger (μ=1.61). The few participants experienced with touchscreen-equipped handhelds had regularly used a stylus for touch input (μ=4.60), but one-handed thumb (μ=2.20) and two-handed index finger (μ=2.00) techniques had been used less often; a two-handed technique using both thumbs had almost never been practiced (μ=1.40).

Hand width and thumb length were recorded for each participant. Thumb length varied between 99 and 125 mm (μ=115 mm, σ = 5.75), and hand width varied between 75 and 97 mm (μ=88 mm, σ = 6.08).

4.2 Equipment

Both phases of the experiment were performed on an HP iPAQ h4155 measuring 7.1 x 1.4 x 11.4 cm with an 8.9 cm screen, measured diagonally. The display resolution was 240x320 pixels with 0.24 mm dot pitch. The study interface and control software was developed using the Piccolo.NET graphics toolkit [1,14].

4.3 Phase 1: Discrete Targets

The goal of the discrete target phase was to determine size recommendations for widgets used for single-target tasks, such as activating buttons, radio buttons and checkboxes.

4.3.1 Design

This phase of the study used a 5 (target sizes) x 9 (locations) x 5 (repetitions) within subjects design. Target sizes were 3.8, 5.8, 7.7, 9.6 and 11.5 mm on each side. We performed pilot studies to determine the appropriate target sizes for the study. Since standard widget sizes range from 2.64 mm (radio buttons) to 4.8 mm (buttons), 3.8 mm represents an average target size for existing devices. Pilot studies indicated that performance rates leveled off for target sizes greater than 11.5 mm and thus represented the largest practical recommended size for singular targets.

Nine target locations were defined by dividing the display into a 3x3 grid of equal-sized cells. For each trial the target was located in the center of one of the nine cells.

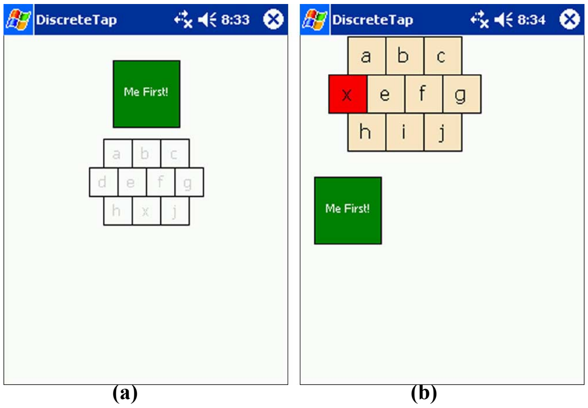
Each target size (5) was tested 5 times in each of the 9 regions for a total of 225 trials. Trials were distributed across 5 blocks. With the first five participants, the sizes and locations of the targets were accidentally randomized across all blocks, but after minor modifications to the software for both phases, the sizes and locations of the targets were randomized within each block to ensure that each size x location combination was tested once per block.

4.3.2 Tasks

The participant's task for each discrete target trial was to tap an initial start position and then the target to be selected. All tasks were performed standing and one-handed, using only the right hand thumb for the interaction with the touchscreen. The participants were instructed to perform the tasks as naturally as they could, favoring accuracy to speed.

For each trial, the start position was indicated by a large green button designed to be easy to select, but from which movement distance could be measured (Figure 1). The distance between the green button and the target was constant for all tasks, while the relative location of the green button varied depending on the region in which the target was positioned. To standardize movement direction across trials, the green start button was located either directly North or South of the target, so chosen because North↔South movement better matches the thumb's natural axis of rotation than East↔West movement. If the target was located in the first row of the grid, the green button was located in the cell below the target. Otherwise, the green button was located in the cell above the target.

Two issues arose in the design of the tap target. First, our pilot studies indicated that lone targets were perceived easier to tap than those near other objects. To address this issue, we



(a) (b)

Figure 1. The experiment interface for the discrete target phase. (a) The startup view for a trial testing a 5.8 mm target in the center zone. (b) The display for a trial in the upper left zone as the user selects the 7.7 mm trial target (x).

surrounded each intended target by 'distractor' targets. This meant participants were required not only to hit a target, but also avoid others. In addition, the design provided an interface closer to real world applications which often present multiple widgets close to each other instead of one single target on the screen. Our second concern was that the constant distance between each start location and target meant that users could conceivably adopt a routine or preprogrammed movement for task completion rather than as a result of explicit aiming. Here, too, the distractor targets were of value. Although the relative position of the target with respect to the start position never changed, the distractors were presented in randomized locations around the target, which promoted a sense that the participant was not moving the same exact distance and in the same direction for each trial.

In each trial, the intended target was designated with an 'x', while the distractors were labeled with other alphabetic characters. At the start of a trial, the target and all distractors were displayed with a white background and light-gray lettering, so as to deemphasize the target, and discourage the locating of the target preattentively before the start of the trial (Figure 1a). When the start button was tapped and released, labels turned black and keys turned pink to draw attention to all on-screen objects (Figure 1b).

Motivated by prior success of the lift-off strategy in touchscreen selection tasks [15] and current use for standard interface widgets of Pocket PC operating systems, the lift-off selection strategy was used in the study. Thus the locations of the participants' on-screen selections were recorded upon thumb release; a successful target selection required that both the tap and release positions were located within the target area. Target taps could also be cancelled by dragging the thumb outside of the target area before the release, similar to the method allowed for canceling widget actions on the Pocket PC.

To ensure visual search was not impacted by the variability of white space surrounding labels as targets changed size, font sizes were scaled with target sizes. Because of limited screen space and evidence that performance is unaffected by key spacing (e.g., [3]), we used 0 mm edge-to-edge spacing between targets and distractors. Participants were provided with both auditory and visual feedback when touching targets. The 'x' target was highlighted in red upon thumb contact (Figure 1b), and both success and error sounds were played upon thumb release to indicate whether the target was hit successfully or not. If a tap was cancelled no auditory feedback was given.

4.3.3 Procedure

The discrete target phase began with a practice session, which consisted of one block of trials: targets were presented once at each size in each location, for a total of 45 trials. After the practice session, users completed the five official trial blocks, followed by a subjective preference questionnaire.

4.3.4 Measures

Application logs recorded the time between the start (first) tap and target (second) tap, the absolute position of the second tap, and trial success or failure. After completing all trials, the participants were asked to rate how comfortable they felt tapping the target 'x' in each region of the screen using a 7-point scale (1 = uncomfortable, 7 = comfortable), as well as which target size was the smallest they felt comfortable using in each region.

4.4 Phase 2: Serial Targets

The goal of the serial target phase was to evaluate required key sizes for widgets used for text or numeric entry.

4.4.1 Design

The serial target phase was a 5 (target sizes) x 4 (locations) x 5 (repetitions) within subjects design. Target sizes were 5.8, 7.7, 9.6, 11.5, and 13.4 mm with 0 mm edge-to-edge spacing. Target sizes were similar to those of the discrete target phase, except due to previous findings that error rates tended to increase for sequential selections [3], the smallest target (3.8 mm) was removed and an even larger target (13.4 mm) added. To study the effect of location on task performance, four regions were defined by dividing the screen into a 2x2 grid.

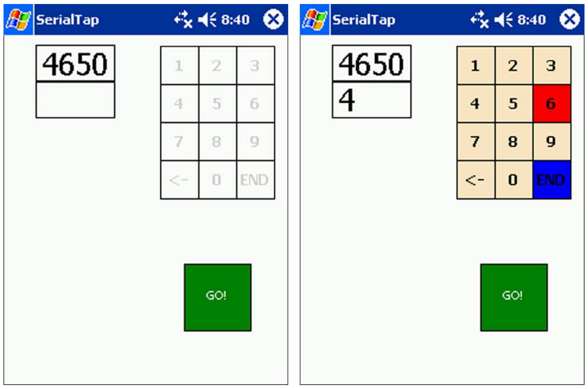
Each of the target sizes (5) were presented 5 times in each of the 4 regions for a total of 100 trials. As in the discrete target phase trials were divided into 5 blocks. Except for the first 5 subjects who received all trials randomized across all 5 blocks, each size x location combination was presented once per block, in randomized order.

4.4.2 Tasks

The serial target task design was based on tasks used for previous studies [2,3]. Subjects were required to enter a series of four digit codes using a soft numeric keypad. They performed the tasks with the thumb of the right hand while standing, as in the discrete target phase.

For each task, a green 'start' button, a numeric keypad and a randomly-generated 4-digit goal sequence were displayed. Backspace and 'END' keys were also presented in the bottom corners of the keypad (Figure 2). Since the keypad's location varied from trial to trial, the remaining interface elements were repositioned as follows: the green 'start' button was positioned in the cell above or below the keypad, and the 4-digit goal sequence appeared to the left or right of the keypad.

The participant's task was to tap the green button first, enter the target sequence with the keypad, and finally touch the 'END' key to confirm the entry and proceed to the next task. The input string was displayed directly below the target sequence. The backspace key could be used for corrections; however it was not necessary



(a) **(b)**

Figure 2. Experiment interface for the serial target phase. (a) The startup view for a trial testing a keypad with 7.7 mm targets in the upper right zone. (b) The display for the same trial as the user selects the second digit in the sequence (6).

for users to input the correct number before moving on - only that they input 4 digits.

Several interaction features were retained from the discrete target phase. After tapping the green ‘start’ button, the background of the keypad changed from white to pink and the labels from light gray to black (Figure 2). Here, too, font sizes adapted to changes in target size. Finally, visual and audio feedback was provided upon target selection. The success sound was played for all key hits, except in the event that the ‘END’ key was selected before all numbers had been entered, or a numeric key was selected after all four digits had been entered; in these cases an error sound was played. Again a lift-off strategy was used for selection.

4.4.3 Measures

Application logs recorded total task time from the release of the start button to the release of the ‘END’ button, the first transition time from the release of the start button to the release of the first keypad button, and the first transition distance. Errors were recorded similarly to Sears et al. [16], where uncorrected errors were recorded by comparing the target and input sequence, and corrected errors by counting the number of backspace sequences. After completing all trials, participants were asked to rate how comfortable they felt using the keypad in each region of the screen using a 7-point scale (1 = uncomfortable, 7 = comfortable), and which keypad size was the smallest they felt comfortable using in each region.

5. RESULTS

5.1 Discrete Target Results

5.1.1 Discrete Task Times

Task time, defined from the release of the start button to the release of the target ‘x’, was analyzed using a 5 x 9 repeated measures analysis of variance (RM-ANOVA) with factors of target size (3.8, 5.8, 7.7, 9.6 and 11.5 mm) and location (9 regions derived from a 3x3 division of the screen). Erroneous trials were eliminated from the data set and the mean total time of the remaining trials was computed. A 5% level of confidence after Greenhouse-Geisser correction was used to determine statistical significance. A main effect of target size, (F(1, 25) = 70.42, p < .001) was observed. No other main effects or interactions were observed.

Not surprisingly, as targets grew in size, participants were able to tap them faster (Figure 3a). Post-hoc comparisons using Bonferroni corrections revealed that time differences between all target sizes were significant, even between the two largest target sizes (p = .04). These results are consistent with Fitts’ Law, which we described earlier. Due to the small screen size and limited practical range of target sizes in this study, the values for task IDs were small, and the range narrow. While these conditions make our study inappropriate for offering official values for *a* and *b*, the Fitts’ model well explains the decrease in tap time with the increase in target size, and hence decrease in task difficulty (Figure 3b).

5.1.2 Discrete Task Percent Error

A 5 x 9 repeated measures analysis of variance (RM-ANOVA) was carried out on the percentage of trials that were performed in error, with factors of target size (3.8, 5.8, 7.7, 9.6 and 11.5 mm) and location (9 regions derived from a 3x3 division of the screen). Once again, a main effect of target size was observed (F(1, 27) = 49.18, p < .001), but no effects of target location nor interactions between target size and location were found.

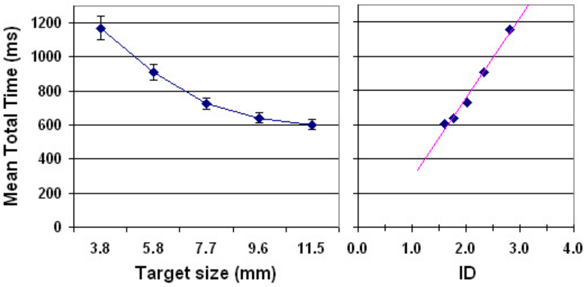


Figure 3. (a) Mean total time between the release of the start button and ‘x’ for each target size in the discrete target phase. (b) Relationship between movement time (MT) and index of difficulty (ID).

As shown in Figure 4, errors declined as target size increased. Post-hoc comparisons using Bonferroni corrections showed that error rates for the two smallest targets differed significantly from one another, and were significantly higher than for all other targets. Also, participants made significantly more errors when aiming for the mid-sized target (7.8 mm) than the largest target (11.5 mm). However, there was no significant difference in error rate between the two largest targets (9.6 v. 11.5 mm). So while speed improves significantly as targets grow from 9.6 mm to 11.5 mm, error rate does not.

5.1.2.1 Discrete Task Hit Distribution

Several investigations into target size requirements have used actual selection location to derive recommendation for on-screen targets. Since error rate was not distinguishable between the two largest targets, Figure 5 displays the on-screen hit distribution for the smallest four targets in all nine screen locations. The nine solid white boxes in each figure indicate the valid hit zones, with the center shown as a black crosshair. Taps that fell within valid bounds are shown as gray dots, and erroneous hits are shown in black. The dark gray outline near each zone center encloses all hits that fell within 2 standard deviations (2-SD) of the means in both the X and Y directions.

Along with each diagram, we present the maximum width and height of any of the 2-SD bounding boxes to offer the minimum sized box that would be expected to enclose 95% of hits at any screen location. We see that in general, the total area of these boxes increases with target size, and thus users are indeed trading off speed for tap accuracy. If we consider the relative shape and

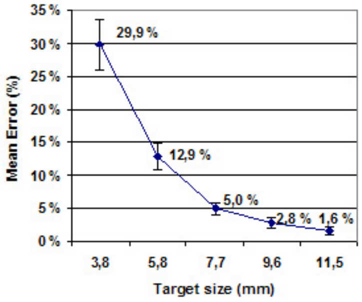


Figure 4. (a) Mean percentage of erroneous trials for each target size in discrete target study phase.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**