

Chucking: A One-Handed Document Sharing Technique

Nabeel Hassan, Md. Mahfuzur Rahman, Pourang Irani, and Peter Graham

Computer Science Department, University of Manitoba
Winnipeg, R3T 2N2, Canada

`nhassan@obsglobal.com, {mahfuz, irani, pgraham}@cs.umanitoba.ca`

Abstract. Usage patterns of private mobile devices are constantly evolving. For example, researchers have recently found that mobile users prefer using their devices with only one hand. Furthermore, current hardware in these devices reduces the need for a stylus and instead relies on finger input. However, current interactive techniques, such as those used for sharing documents between private and public devices have not taken advantage of these recent developments. For example a popular technique, Flick for sharing documents between devices relies on pen and stylus use and has not been adapted to support one-handed interaction. In this paper, we present Chucking, a gesture-based technique for sharing documents between private mobile devices and public displays. Chucking is based on the natural human gesture used for throwing or passing objects. We present the various design parameters that make Chucking an effective document sharing technique. In a document positioning task, we evaluated Chucking against Flicking. Our results show that under certain contexts users were more accurate and effective with Chucking. Participants also preferred Chucking as it maps closely the type of interaction one naturally performs to share objects. We also introduce extensions to Chucking, such as Chuck-Back, Chuck-and-Rotate, and Chuck-and-Place that constitute a suite of techniques that facilitate a large range of document sharing interactions between private mobile devices and public displays.

Keywords: Chucking, Flicking, public-to-private document sharing, multi-document environments (MDEs).

1 Introduction

We are witnessing the introduction of public displays in numerous environments, such as in schools, airports, museums, and shopping centers. The recent proliferation of such displays has led to the establishment of multi-display environments (MDEs) [1,2,3,14] in which several private devices (PDAs, cell-phones, etc) can be used to interact with information available on public displays. Recently, there has been growing interest in determining ways to bridge the interactions that couple together private and public displays [12,22,25]. To this end, one common application that has emerged is that of sharing documents between devices. This includes scenarios such as an instructor sharing photographs from a field trip with a classroom, a group of users sharing their private documents in a boardroom during a meeting or organization members placing information on a bulletin board like display.

Researchers have developed and studied document sharing techniques across multiple devices and platforms [5,12,21]. While such techniques have shown significant advantages in specific usage contexts, they require the use of a stylus or pen to facilitate the interaction, a feature that is becoming less common on platforms such as the iPhone and smart phones. Furthermore, current designs do not easily support one-handed interactions, a mode of operation that is highly favored when using mobile handheld devices [9,11]. Generally, for the task of sharing documents from a private device to a public display, no current method is able to satisfy all of the following design goals:

- *Stylus independence*: on multiple devices, styli operation is non-existent or limited. Document sharing should not be inhibited by the lack of external styli.
- *One handedness*: studies reveal that 74% of mobile device users use one hand when interacting with their devices [9,10]. New mobile applications should support the prevalent use of one-handed operations.
- *Natural and fluid*: techniques that map to natural ways of working are easier to learn so any new design should be as natural as possible.
- *Position independence*: one-handed interaction suffers from a non-uniform distribution of thumb reach. Limitations of thumb reach should not prevent a technique from operating uniformly well across different conditions.

We propose Chucking, a one-handed gesture-based interaction for sharing documents from a private mobile device to a public display (Figure 1). Chucking was designed to alleviate some of the limitations of existing techniques for one-handed use on devices that do not have a stylus. In a document positioning task, users were more accurate with Chucking than one-handed Flicking. Furthermore, participants preferred Chucking to Flicking and found the technique natural to learn and use. We also introduce a suite of techniques based on Chucking, including Chuck-Back for pulling documents from the public display to the private device, Chuck-and-Rotate to rotate a chucked document to accommodate orientation issues on tabletops, and Chuck-and-Place, a technique to allow for more accurate document placement.

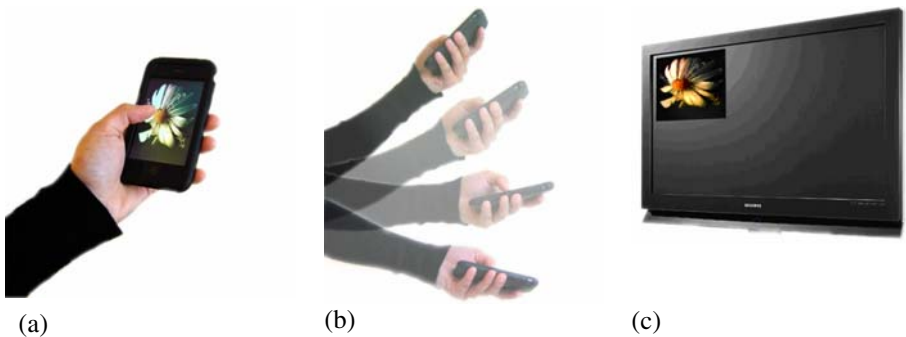


Fig. 1. From (a) a steady state, (b) the user Chucks a document to the public display (c) with an extension of the forearm or wrist. The gesture is analogous to dealing cards like a dealer.

2 Related Work

We present the related work on one-handed interactions, on sharing documents in multi-display environments, and on tilt-based interactions.

2.1 One-Handed Interaction

One-handed interaction is a highly popular method for interacting with mobile devices. Karlson and Berderson, conducted several in-situ observations which concluded that 74% of mobile users employ one hand when interacting with their cellular devices [10]. The same observations and web surveys suggested that PDA users were inclined to more often use two hands, but users of such devices also expressed significant interest in using one hand if possible, suggesting the need for better tools to support one-handed interactions.

Applens and LaunchTile were two of the earliest systems to support one-handed thumb use on PDAs and cell-phones [11]. With Applens, users were provided a simple gesture set, mimicking the up-down-left-right keys, to navigate a grid of data values. LaunchTile provided access into the tabular data view by allowing users to press on soft buttons associated with an area of the grid. In a user study, users performed correct gestures 87% of the time, suggesting that simple thumb gestures are memorable. However, they also found that users were reluctant to use gestures and preferred tapping. Their results also showed that error rates were also influenced by the direction of the gestures and the number of gestures available, suggesting a limit on the number of different gestures one should design.

Further investigation by Karlson et al [10] on the biomechanical limitations of one-handed thumb use has revealed that users do not interact with all areas of a device with equal facility. Instead, user grip, hand size, device ergonomics, and finger dexterity can influence movement and thumb reach. For example, right handed users have limited thumb movement in the NorthEast-SouthWest direction. Additionally, regions of the device away from the right edge are more difficult to reach. These findings supported the development of Thumbspace [9]. To facilitate thumb reach with Thumbspace, users customize and shrink the entire workspace into a 'thumb-reachable' bounded box. In an extensive study, users performed better at selecting targets that would be further to reach, using Thumbspace than other techniques. Used in conjunction with Shift [23], a technique for high-precision target selection with the finger, Thumbspace resulted in higher accuracy than using either technique alone.

Overall, the results of these studies suggest that while there is sufficient evidence of one-handed use with mobile devices, we do not have a broad range of techniques to support such usage. Furthermore, the design of one-handed interactions needs to be concerned with the size of object, the placement and position of control items, and the range of allowable gestures with the thumb. We considered these limitations in the design of Chucking.

2.2 Document Sharing Techniques

Researchers have proposed a number of techniques to support direct document sharing between private and public devices [21,27,12,1,2]. Pick-and-drop [21] is a

technique that allows a user to tap on an object with a digital pen to ‘pick it up’ and then to ‘drop’ the object in another location on the workspace. However, pick-and-drop does not support very well cases when the user is interacting with a very large display unless the user can directly reach the desired object. To resolve this ‘out-of-reach’ dilemma, researchers have proposed proxy based techniques to allow the user to interact with distant content. In the vacuum filter [2] and drag-and-pop [1] techniques, users pull in proxies of remote content, with a brief gesture of the stylus. In this manner, users can either select distant objects or drag-and-drop items without carrying out large arm movements or displacing themselves physically. However, with both the vacuum filter and drag-and-pop approaches, users are required to be in close proximity to the large display, a situation that is inappropriate in certain contexts such as boardroom meetings.

Researchers have also proposed document sharing techniques that do not require strict proximity with the display [19]. Most techniques in this group have investigated the use of human gestures for resolving issues with distance [3,5,27]. The use of human gestures in interactive systems has a natural appeal as designers believe that such gestures are analogous to how we interact with physical objects in the real world. Additionally, gestures can be efficient if they are designed to integrate both command and operand in a single motion, to conserve space on devices, and to reduce the need for buttons and menus [9].

Toss-it is a gesture-based technique to facilitate data transfer between mobile devices [27]. With this technique, users can ‘toss’ information from one PDA to another with a swinging action, analogous to pitching a softball. Based on the ‘strength’ of the action and the layout of mobile devices in the environment, toss-it selects the device that is most suited to receive the data. Toss-it inspired the design of Chuck. However, toss-it allows users only a limited number of gestures. Furthermore, it has not been designed to work with large public displays, thus issues such as positioning or orientation of documents are not easily resolved using toss-it. Finally, toss-it does not exploit the full range of gestures possible by the human hand.

Researchers have also proposed techniques that are slightly more natural than toss-it. Geißler’s throw technique [3] requires the user to make a short stroke over a document, in the direction opposite to the intended target, followed by a long stroke in the direction of the target. Wu et al. [26] describe a ‘flick and catch’ technique, in which an object is ‘thrown’ once it is dragged up to a certain speed. Reetz et al. [19] demonstrated the benefits of Flicking as a method for passing documents over large surfaces. Flicking was designed to mimic the action of sliding documents over a table, and closely resembles the push-and-throw model designed by Hascoet [5]. Flicking was found to be much faster than other document passing techniques for tabletop systems [13] but incurred a noticeable cost in terms of accuracy. Superflick [19] was designed to improve the accuracy of flick by allowing the user to adjust the position of the object on the shared display. Superflick performed with equal accuracy and as efficiently as radar, a technique that provides an overview of the public space on the user’s private device [19]. While radar is a fairly robust technique, the overview space it covers would be impractical on private devices such as mobile displays, where issues such as occlusion and small targets would make radar unusable.

Flicking has numerous advantages for various interactions. However this technique has not been evaluated under conditions of mobility where the usage is one-handed

and on devices without a stylus. Furthermore, most of the document sharing techniques discussed above are not designed for bidirectional transfer (i.e. back and forth). We present variants of chucking that allow for bidirectional transfer, for accurate positioning and for rotating objects on horizontal public displays (section 7).

2.3 Tilt-Based Techniques

Chucking builds on tilt-based techniques. One of the earliest uses of tilt input was proposed by Rekimoto [20] for invoking an analog input stream where tilt could be used to invoke menus, interact with scroll bars, or to browse maps. One particularly appealing feature of such an interaction, as noted by Rekimoto, was the use of only one hand to both, hold and operate the device. Since Rekimoto's proposal, a significant number of tilt-based techniques have emerged. Recent work in tilt interfaces can be grouped into rate-based or position-based tilting. Hinckley et al. [6] demonstrated the use of accelerometer enabled tilting for automatic screen orientation and scrolling applications, a now common feature on commercial devices.

Numerous systems have defined a fixed mapping from tilt position to a function in the workspace. Oakley and Park [17] described a tilt-based system with tactile augmentation for menu navigation. Each menu item was selectable by tilting the device at a fixed angle. TiltText [24] allows text entry by tilting the device to the correct position to enter a letter. Results have shown that this form of interaction can improve text entry speed but at the cost of errors. A study by Oakley and O'Mondrain [16] has shown that position-based mapping is more accurate than rate-based mapping, presumably due to the feedback obtained based on the position of the arm. None of the above described systems have fully explored tilt for one-handed document sharing.

3 Design Framework

We propose a framework for chuck-based interactions. This framework highlights five primary factors that influence performance with chucking: gesture commands, mapping function, feedback, invocation and simultaneous touch input.

3.1 Gesture Commands

Chucking is a one-handed document sharing interaction. To invoke a Chuck, the user selects an item with the thumb or other input (e.g. jog dial) and then extends the forearm or wrist as in chucking cards on a table. The use of a hand gesture alleviates the need for a stylus or pen. To recognize the gesture, we use the values from tilt sensors, a common feature on many devices such as the iPhone or smartphones. With 2D tilt sensors, we are capable of allowing users a more flexible range of gestures that are natural to the concept of throwing to share documents [5].

3.2 Mapping Function

In Chucking we use a mapping function to position the document in a given location on the remote display. Existing proposals for tilt-based interactions have suggested a mapping using either the rate of tilt or the angle of tilt to control a virtual cursor. In a rate-based tilt design, the speed of the cursor is mapped to the degree of tilt. Designers

have reported that rate-based systems are difficult to control and do not provide the level of precision possible with position-based control. In a study, Oakley and O'Modharin [16] found that users were more efficient and more accurate controlling items on a menu with position-control than rate-control tilting. They observed that a primary benefit of position-based tilt control is the lack of reliance on any hidden virtual model. Additionally, position-based tilt reinforces feedback with the physical orientation of the hand. We designed Chucking based on position-control mapping instead of rate-based mapping. Several variants of position-based mappings exist. An absolute position mapping captures the absolute tilt value and uses this to identify the direction of a user's chuck. Alternatively, we can also use a relative position mapping, which maps the amount of tilt to the position of the document on the remote display. We applied a mapping in which shorter differences in orientation are mapped to closer locations on the remote display, while larger movements are mapped to locations that are further away from the user.

3.3 Feedback

A feedback loop is a necessary component of many interactive techniques. In the case of tilt, the user can infer a certain degree of feedback from the position of the hand. However, such a feedback mechanism is insufficient if the interaction technique necessitates a large number of unique tilt gestures. Visual feedback is a common feedback method for most tilt-based techniques [16,17,20,24]. With visual feedback, the tilt techniques are restricted to the range of motion such that the feedback signal is not inhibited. Auditory feedback can also complement visual feedback for tilting [17]. However, this form of feedback is limited based on the context of its use and the specific techniques employed. Oakley and Park [17] presented the case for motion based interaction without any visual feedback but with limited tactile feedback. To evaluate if their systems could be used eyes-free, they gave participants very minimal training with their menu system. Their results showed that performance improved with visual feedback and accuracy was unaffected in the eyes-free condition. Based on prior work, complete visual feedback, in which clear visual indication suggests when the transfer has occurred, would benefit most Chucking based interactions. However, augmenting this form of feedback with audio also enhances the interaction and demarcates the completion of the task. We use both visual and auditory feedback. Auditory feedback consisted of a 'click' sound when the transfer is successful.

3.4 Invocation

Since chucking is based on tilt interactions, the system needs to disambiguate typical non-chuck gestures, with those that are intentionally created for the purpose of sharing documents. One method for invoking the gesture is to simply click anywhere on the device, and then to start the gesture. Alternatively on devices that are not touch-based, the user can press a button or the jog dial to select the item to transfer.

3.5 Sequential Touch Input

Since Chucking relieves the fingers from performing any major work, we can take advantage of this and combine the use of fingers in the interaction. For document

positioning two possible configurations are available. By embedding a touch mechanism with Chucking, users can place their finger in one of several locations before sending the document (i.e. in conjunction with the velocity vector) to fine tune adjustments such as in Superflick [19]. Alternatively, the finger can be used for orienting the document on a horizontal display. The use of a finger position on the private display can suggest the orientation the document may take once it is placed on the horizontal surface. We discuss these details in section 7 when presenting Chuck-and-Rotate, and Chuck-and-Place.

4 Data Capture

We first captured data points, consisting of angular tilt in the X and Y dimensions of the 2D tilt sensor, to identify distinct sets of gestures for Chucking. We used a Dell AximX30 PDA with a TiltControl accelerometer attached to it and polled the X and Y tilt angles of the TiltControl at 40 msec intervals. The application was written in C#.NET. Four right-handed volunteers performed a set of gestures. All participants were university students; three males and one female (average age of 23.25 years). All had experience using PDAs but none had any exposure to the Chucking technique.

4.1 Method for Capturing the Unique Gestures

Participants were asked to perform the Chucking gesture as if they would be tossing an object in one of nine different locations on a grid: Right-High, Right-Medium, Right-Low, Center-High, Center-Medium, Center-Low, Left-High, Left-Medium and Left-Low. We limited the data capture to only nine locations as pilot studies revealed significant overlap of tilt values for anything larger than a 3×3 grid. Additionally, the basic Chucking technique is suitable for sending documents in a general region, and we provide refined control for accurate positioning with Chuck-and-Place (section 7).

After an initial demonstration, the participants were asked to chuck an imaginary item on the PDA for a total of five trials for each condition. The experiment collected a total 180 trials as follows:

$$4 \text{ participants} \times 5 \text{ trials per gesture} \times 9 \text{ gesture locations} = 180 \text{ trials.}$$

The application recorded sample points from the moment the users pressed the 'Capture' button with the thumb to when they released it after completing the gesture. The user repeated the same gesture for a total of five readings and then the user was asked to perform another five gestures for another location on the grid.

4.2 Data Analysis

We used the X and Y angular readings to find the average relative displacements that created unique gestures. We found that the distinct cutoff points in the X movement were unambiguous. However, average tilt displacements in the Y direction varied and standard deviations revealed some overlap among the set of High/Medium/Low gesture locations.

Based on the samples collected we found that for right sided gestures, the average X angular points was clearly positive for gestures toward the right and negative for left gestures (Figure 2.a). To identify distinct relative movement for High, Medium or

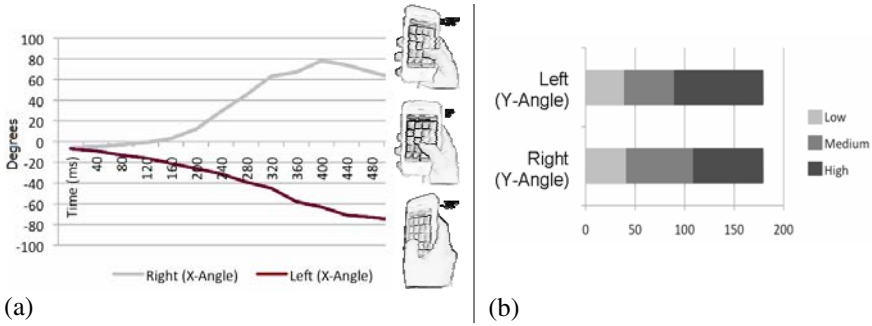


Fig. 2. (a) As can be seen from the above sample of X-Angle values, the X-Angle average is positive for Right gestures (+30° average) and negative for Left gestures (-30° average); (b) High, Medium and Low cutoff points for right/left sides have distinct ranges and are somewhat different due to differences in wrist ulnar/radial deviation for left and right sides [276]

Low positions, we inspected the Y angular values. Where unlike the right-left gestures that revealed sufficiently distinct averages on the X-axis of tilt, the distinct ranges for the relative angular movements in the Y-axis varied based on whether the user was Chucking to the Left or to the Right (Figure 2.b).

4.3 Determining the Gestures

X and Y angular points reveal unique patterns which help distinguish what gesture the user is performing with Chuck. For example we know that if the average of X is greater than zero, a right-sided gesture is being carried out. To break it down further, we use the distance in Y to determine if Chucking is to be mapped to a High, Medium or a Low gesture. The distance was calculated as: $\sqrt{(\Omega_x - \Omega_y)^2}$ (where Ω is the Y-angle).

This form of relative mapping of X and Y angular positions was found to be easier to learn and less error prone as opposed to having an absolute mapping where gestures had to fall within a fixed range of angles to be recognized successfully.

5 Experiment and Results

We carried out an experiment to assess the limitations and performance of Chucking. We evaluated Chucking against Flick, with a target positioning task adapted from [14]. We also collected subjective ratings from the participants.

5.1 Hardware Configuration

We used the same PDA as described in section 4. The public display was simulated using a 52 inch monitor connected to a PC, to which the users transferred the targets. The PDA and PC communicated via a wireless connection and all the data was collected on the PC.

5.2 Performance Measures

The experimental software recorded number of errors, number of attempts, failure rate, and completion time as dependent variables. We recorded an *error* on any given trial in which the user was unable to move the PDA object to the *correct* target location. The trial ended only when the user selected the correct target, so multiple errors or multiple *attempts* were possible for each trial. For each trial we recorded the number of attempts used in correctly transferring the object. After five attempts, we marked the trial as a *failed trial*, and the participant then started the next trial. Trial completion time is defined as the total time taken for the user to either Flick or Chuck the object successfully. Since arm movements take longer and have a wider range-of-motion than finger movements, completion times with Flicking should be lower. However, we expected error and failures rates to be lower with Chucking as the user can engage in wider ranges of movements with the arm, than with the thumb.

5.3 Participants

Ten participants (8 males and 2 females) between the ages of 20 and 30 were recruited from a local university. All subjects had previous experience with graphical interfaces and were right-handed. None of the users were color blind, thus allowing the users to clearly see the red target objects. Furthermore, all ten participants were familiar with tilt based interactions, either from using the Wii™ or the iPhone™.

5.4 Task and Stimuli

To test whether participants could accurately and effectively position an object from the PDA onto the large display, we devised a task that consisted of placing a PDA object onto a target location on the large surface. The target location on the large surface was in one of several positions in a grid. Targets on the PDA were positioned in one or two locations, either at the center or toward the edge of the PDA. The choice of placement is representative of conditions in which the user would necessitate reaching one of the PDA locations with the thumb when operating with one-hand. Furthermore, this simulates the common case of having objects in an image browsing application, or any other application in which a list of items typically appear as thumbnails or icons. The object on the target position on the large surface appeared in red. If the user moved the PDA target to an inaccurate surface position, that position would get highlighted in yellow. When the object landed accurately on the surface target, the target changed color to green, and the user was then presented with the next trial. A trial would not be completed unless the user accurately moved the document to the desired position on the large surface. Users who failed the first time were given five attempts, after which the trial was marked as failed. The position to transfer to was randomly selected on the large surface as well as the target object on the PDA.

5.5 Procedure and Design

The experiment used a within-participants factorial design. The factors were:

- Technique: Flick, Chuck.
- Location (of object on PDA): Center, Edge
- Grid Size: 2×2, 3×2.

The order of presentation first controlled for technique. Half the participants performed the experiment first with Flick and then with Chuck. Levels of all the other factors were presented randomly. We explained the task and participants were given ample time to practice the techniques with various parameters of the other independent variables. The experiment consisted of two blocks with each block comprising ten trials per condition. With 10 participants, 2 Techniques, 2 Locations, 2 Grid Sizes (4 or 6 positions), the system recorded a total of $(10 \times 2 \times 2 \times 4 \times 2 \times 5) + (10 \times 2 \times 2 \times 6 \times 2 \times 5) = 4000$ trials. The experiment took approximately 45 minutes per participant, including training time.

6 Results

We recorded a total of 4000 trials. We present our results in terms of number of error trials, number of attempts, number of failures, and completion times.

6.1 Error Rate

We recorded an error on any given trial in which the user was unable to Chuck the PDA object to the correct location on the remote display. We used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) for all our analyses with subjects as random factors.

There was a main significant effect of *Technique* ($F_{1,9}=38.295$, $p < 0.001$) and of *Grid Size* ($F_{1,9}=71.303$, $p < 0.001$) on error rate. Surprisingly, we did not find any main effects for *Location* ($F_{1,9}=0.242$, $p = 0.634$) on error rate. We found no significant interaction effect between these factors ($F_{1,9}=0.698$, $p = 0.425$, for *Technique* \times *Grid Size*), ($F_{1,9}=3.649$, $p = 0.088$, for *Technique* \times *Location*), ($F_{1,9}=0.013$, $p = 0.911$, for *Grid Size* \times *Location*). From the 4000 experimental trials (split in half for each technique), users performed an error on 342 trials with Chuck (over 2000 trials), yielding an error rate of 17.1% and 686 errors with Flick (over 2000 trials), resulting in an error rate of 34.3%. Figure 3.a shows the average error rate for each Technique and Location by Grid Size.

6.2 Number of Attempts

For each error trial, users could retransfer the target up to a limit of five attempts. After five attempts, we marked the trial as a *failed trial*, and the participant then started the next trial. Since users were asked to retry after an error, we also recorded the number of attempts taken to position a target accurately, and analyzed this dependent variable. There was a main significant effect of *Technique* ($F_{1,9}=59.395$, $p < 0.001$) and *Grid Size* ($F_{1,9}=16.675$, $p < 0.01$) on number of attempts. We did not find any main effects for *Location* ($F_{1,9}=2.097$, $p = 0.182$) on number of attempts. We found no significant interaction effect between *Technique* \times *Grid Size* ($F_{1,9}=0.001$, $p = 0.921$) and *Grid Size* \times *Location* ($F_{1,9}=.114$, $p = 0.743$). We found an interaction effect for *Technique* \times *Location* ($F_{1,9}=6.565$, $p = 0.031$). Overall participants performed an average of 1.29 (s.d. .79) attempts with Chuck and 1.75 (s.d. 1.3) attempts with Flick. Figure 3.b shows the average number of attempts for each Technique and Location by Grid Size.

6.3 Failure Rate

After five attempts, we ended the trial and recorded it as a failure. There was a main significant effect of *Technique* ($F_{1,9}=14.865, p = 0.004$). We did not find any main effect for *Grid Size* ($F_{1,9}=1.186, p = 0.304$) or *Location* ($F_{1,9}=3.154, p = 0.109$) on the average number of failures. We found no significant interaction effect between *Technique* \times *Grid Size* ($F_{1,9}=0.966, p = 0.351$), *Grid Size* \times *Location* ($F_{1,9}=2.111, p = 0.180$) and *Technique* \times *Location* ($F_{1,9}=2.727, p = 0.133$). Overall participants failed on 20 trials with Chuck (for a failure rate of 1%) and on 88 trials with Flick (resulting in a failure rate of 4.4%). Figure 3.c shows the average failure rate for each Technique and Location by Grid Size.

6.4 Completion Times

We measured the completion time for each successful trial. With Flick the completion time was the time from the moment the user placed his/her finger on the display up to the lift-off of the Flick. With Chuck, completion time elapsed from touching the target to the time to lift-off the finger to signal the completion of the gesture.

There was a main significant effect of *Technique* ($F_{1,9}=29.121, p < 0.001$) and of *Grid Size* ($F_{1,9}=14.285, p < 0.001$) on completion time but no main effects for *Location* ($F_{1,9}=0.412, p = 0.537$). We found no significant interaction effect between *Technique* \times *Grid Size* ($F_{1,9}=4.967, p = 0.053$), *Grid Size* \times *Location* ($F_{1,9}=0.176, p = 0.685$), or *Technique* \times *Location* ($F_{1,9}=0.534, p = 0.484$). On average it took participants 777 msecs to complete a trial with Chuck and 352 msecs with Flick. Figure 3.d shows the average completion time for each Technique and Location by Grid Size. It

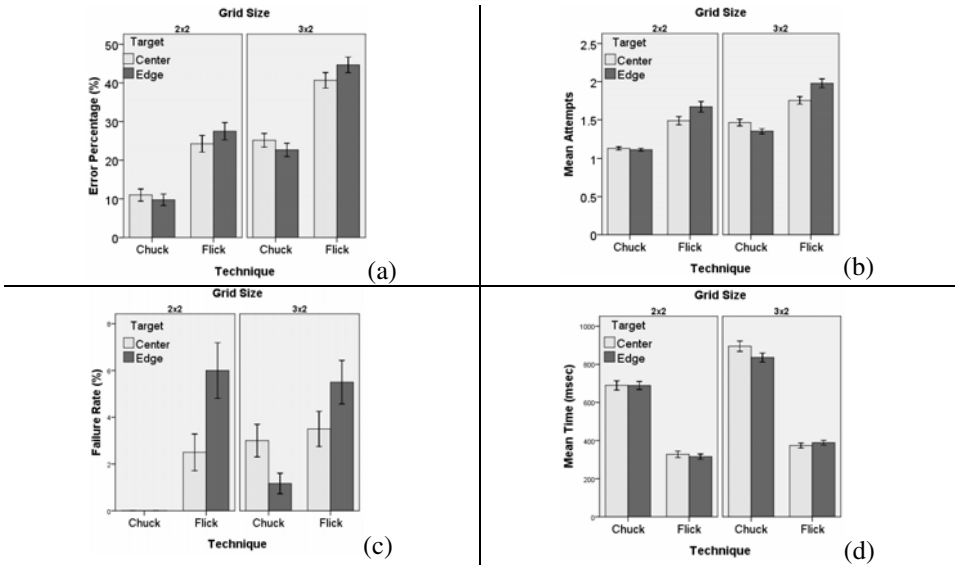


Fig. 3. (a) Average error rate, (b) Average number of attempts, (c) Average failure rate (d) Average completion time; grouped by Target Location and Grid Size. (+/- 1 s.e.)

is not surprising that flicking took users less time to complete. The length of the gesture is significantly less with flick and finger movements are known to take much less time than full arm movements [4].

7 Extensions of Chucking

We designed three extensions to Chucking to create a suite of techniques for coupling private devices with public displays. These are primarily focused on achieving some of the primary tasks for document sharing, including document retrieval, precise document placement and document rotation on the public display. Very few if any analogies to these techniques exist with current systems (e.g. Toss-it [27] or Flick [13]). We demonstrate a complete set of private-public coupling interactions with Chuck-based metaphors.

7.1 Chuck-Back, Chuck-and-Place, Chuck-and-Rotate

While researchers have given significant attention to moving documents from the private device onto the public display, less attention has been placed on the opposite transfer. In *Chuck-Back*, users are able to retrieve objects from the public display and place them on the private device. To invoke Chuck-Back, users press the jog dial on the device. This places a cursor on the public display. The user can control the movement of the cursor by tilting the device forward, backward or side-to-side. Upon hovering over an object, the user can then initiate a pullback by making a motion in the direction opposite that of chucking (a flexion of the forearm). When the user releases the jog-dial, the intended public document is placed on the private device.

Chucking can also be extended to facilitate accurate object positioning. For experimental purposes, we restricted our document sharing task to only a few segments of the available space on the remote display. Similar to Superflick [14], *Chuck-and-Place* allows users to accurately position the document anywhere in the xy -plane, regardless of a grid. Unlike Superflick, we allow for fluid and accurate positioning by allowing the user to perform slight wrist movements within a given time period after transferring the document. Forward and backward tilt, moves the object in the y -plane, while side-to-side tilt moves the object in the x -plane.

Researchers have studied specific techniques to facilitate the orientation of documents on public spaces, such as on tabletops. Rotate-and-Translate (RNT) [14], allows the user to position a document and at the same time rotate it in the correct orientation once passed over to another side of the table. Similarly, *Chuck-and-Rotate* allows the user to freely rotate the document either in a clockwise or counter-clockwise manner using the thumb once the transfer is complete. We map the rotation of the thumb directly onto document rotation such that a thumb motion on the private device also rotates the active object on the remote display.

8 Discussion

We now provide a general discussion of our results and describe some potential limitations of Chucking.

8.1 Performance and Limitations of Chucking

Our results reveal that users are highly accurate with Chucking when the technique is used for transferring objects to general areas of a display. In particular, Chucking alleviates the common problems associated with thumb reach in one-handed interaction. As a result, this technique can reduce steps that would normally be required to bring objects closer to the thumb. Our results, although not statistically significant, show that target location of the object on the PDA was easier to select with Chucking when the object was more accessible (i.e. at the center).

We also observe that performance with either technique was better with a smaller size grid than a larger grid. This suggests that there is a threshold grid size beyond which performance will start to degrade. This is generally expected since there is only a limited range-of-motion for the wrist and forearm that will result in distinct gestures [14]. Chucking is particularly useful in cases where devices such as a cell phone or an iPod do not provide support for pen input. Instead, many of these devices use tilt sensors so Chucking would adapt well to these environments. Additionally, researchers have found that for brief interactions user prefer to not remove the stylus on a device and instead to use their fingers or hands. Chucking would be particularly beneficial for these types of situations. Finally, our study was only conducted with right-handed users. As such the mappings we used are specific to this group of individuals. Generally, ranges-of-motion for the wrist and forearm are symmetrical across both hands [14]. Thus, with minor modifications to our application, Chucking could work for left-handed users. We will investigate the necessary changes to make Chucking adapt to handedness.

8.2 Recommendations

Based on the results and observations we provide the following recommendations for creating one-handed techniques for document sharing:

- One-handed tilt gestures for document sharing are effective for mobile devices, particularly on those devices that lack pen input but provide built-in tilt sensors.
- Performing hand gestures can relieve the fingers for other tasks that may be required during the interaction, such as to rotate an object in a given direction.
- Hand gestures appear natural to users and can be learned with little effort if developed with the appropriate mappings, such as the use of relative tilt input.

9 Conclusion

We introduce Chucking, a one-handed tilt based technique that uses natural metaphors to assist users to share documents from a private device to a public display. In a study we show that users are more accurate with Chucking than one-handed Flicking. Chucking also provides the opportunity to map other tasks (such as orienting a document during the transfer) using the added input dimension of fingers such as the thumb, during the interaction. Any finger based technique for this type of task is inhibited by the range of coverage that the finger has on the device and by the edges of the device. We also introduce extensions to Chucking to create a suite of techniques that facilitate a range of interactions between mobile devices and public displays.

Additional work is needed to investigate thoroughly the simultaneous use of fingers and arm movements in one-handed interaction, the achievable levels of precision possible with Chucking, and the use of Chucking for other tasks, such as map navigation or document browsing.

References

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., Zierlinger, A.: Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In: Proc. of Interact 2003, pp. 57–64 (2003)
2. Bezerianos, A., Balakrishnan, R.: The vacuum: facilitating the manipulation of distant objects. In: Proc. of CHI 2005, pp. 361–370 (2005)
3. Geißler, J.: Shuffle, throw or take it! working efficiently with an interactive wall. In: Proc. of CHI 1998 Extended Abstracts, pp. 265–266 (1998)
4. Grandjean, E.: Fitting the task to the man: an ergonomic approach, p. 372. Taylor & Francis, Abington (1969)
5. Hascoët, M.: Throwing models for large displays. In: Proc. of British HCI 2003, pp. 73–77 (2003)
6. Hinckley, K., Ramos, G., Guimbretière, F., Baudisch, P., Smith, M.: Stitching: pen gestures that span multiple displays. In: Proc. on Advanced Visual Interfaces (AVI 2004), pp. 23–31 (2004)
7. Hinrichs, U., Carpendale, S., Scott, S.D.: Evaluating the effects of fluid interface components on tabletop collaboration. In: Proc. on Advanced Visual Interfaces (AVI 2006), pp. 27–34 (2006)
8. Kruger, R., Carpendale, S., Scott, S.D., Tang, A.: Fluid integration of rotation and translation. In: Proc. of CHI 2005, pp. 601–610 (2005)
9. Karlson, A., Bederson, B.: One-handed touchscreen input for legacy applications. In: Proc. CHI 2008, pp. 1399–1408 (2008)
10. Karlson, A., Bederson, B., Contreras-Vidal, J.: Understanding Single-Handed Mobile Device Interaction. Tech. Report HCIL 2006 (2006)
11. Karlson, A., Bederson, B., SanGiovanni, J.: Applens and launchtile: two designs for one-handed thumb use on small devices. In: Proc. CHI 2005, pp. 201–210 (2005)
12. Maunder, A., Marsden, G., Harper, R.: SnapAndGrab: accessing and sharing contextual multi-media content using bluetooth enabled camera phones and large situated displays. In: Proc. CHI Extended Abstracts (CHI 2008), pp. 2319–2324 (2008)
13. Nacenta, M., Aliakseyeu, D., Subramanian, S., Gutwin, C.: A comparison of techniques for multi-display reaching. In: Proc. of CHI 2005, pp. 371–380 (2005)
14. Nacenta, M., Sallam, S., Champoux, B., Subramanian, S., Gutwin, C.: Perspective cursor: perspective-based interaction for multi-display environments. In: Proc. of CHI 2006, pp. 289–298 (2006)
15. Nacenta, M., Sakurai, S., Yamaguchi, T., Miki, Y., Itoh, Y., Kitamura, Y., Subramanian, S., Gutwin, C.: E-conic: a perspective-aware interface for multi-display environments. In: Proc. of UIST 2007, pp. 279–288 (2007)
16. Oakley, I., O’Modhrain, M.: Tilt to scroll: Evaluating a motion based vibrotactile mobile interface. In: Proc. WHC 2005, pp. 40–49 (2005)
17. Oakley, I., Park, J.: A motion-based marking menu system. In: Proc. CHI 2007 Extended Abstracts, pp. 2597–2602 (2007)

18. Pering, T., Anokwa, Y., Want, R.: Gesture connect: facilitating tangible interaction with a flick of the wrist. In: Proc. of Tangible and Embedded Interaction 2007, pp. 259–262 (2007)
19. Reetz, A., Gutwin, C., Stach, T., Nacenta, M., Subramanian, S.: Superflick: a natural and efficient technique for long-distance object placement on digital tables. In: Proc. Graphics Interface 2006, pp. 163–170 (2006)
20. Rekimoto, J.: Tilting operations for small screen interfaces. In: Proc. UIST 1996, pp. 167–168 (1996)
21. Rekimoto, J.: Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In: Proc. UIST 2007, pp. 31–39 (1997)
22. Swindells, C., Inkpen, K., Dill, J., Tory, M.: That one there! Pointing to establish device identity. In: Proc. of UIST 2002, pp. 151–160 (2002)
23. Vogel, D., Baudisch, P.: Shift: a technique for operating pen-based interfaces using touch. In: Proc. of CHI 2007, pp. 657–666 (2007)
24. Wigdor, D., Balakrishnan, R.: TiltText: Using tilt for text input to mobile phones. In: Proc. of UIST 2003, pp. 81–90 (2003)
25. Wilson, A., Sarin, R.: BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In: Proc. of Graphics Interface 2007, pp. 119–125 (2007)
26. Wu, M., Balakrishnan, R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proc. of UIST 2003, pp. 193–202 (2003)
27. Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., Hashizume, H.: Toss-it: intuitive information transfer techniques for mobile devices. In: Proc. CHI Extended Abstracts 2005, pp. 1881–1884 (2005)