Text input mechanisms for devices with backside touch input

Author 1
Affiliation
Affiliation
author@a.com

Author 2
Affiliation
Affiliation
author2@b.com

ABSTRACT
Talk about text input being a problem for foot-scale devices. The paper explains a few mechanisms that would mitigate that. Discusses the exeriment gains, results both qualitative and quantitative.

Author Keywords
QWERTY, multitouch, touchscreens, keystroke per character(KSPC), words per minute(WPM)

ACM Classification Keywords
H.5.2 Interaction techniques and devices: Miscellaneous—*Optional sub-category*

INTRODUCTION
With an estimated 4 billion units in use in December 2008, mobile devices have already become the most popular computing device in human history. Their portability and communication capabilities have revolutionized how people interact with each other. Despite the rapid growth of mobile phones, text entry on mobile devices remains a major challenge. However, with the recent rise in sales in the smartphone/PDA market and the recent launch of products like iPad, users have greater access to medium sized touch screens. These screens should offer more screen space than a traditional phone, and can arguably be more effective in text-input tasks. However, due to increased size and weight, these devices require specific postures from the users to be able to input text.

Recently, there has been a push in the industry for devices with backside input. Notion ink's Adam PC has a backside trackpad [Reference]. Samsung and Toshiba have also been making effort towards backside touch input [Reference]. With the growing interest in backside touch input, comes the potential of using the same for manipulating information. A ”natural” method of holding such a device is to have hands on both the sides with fingers wrapped around on the back [Figure]. The advantage of having a backside touch input is that users can possibly use the fingers on the back to input text and manipulate information.

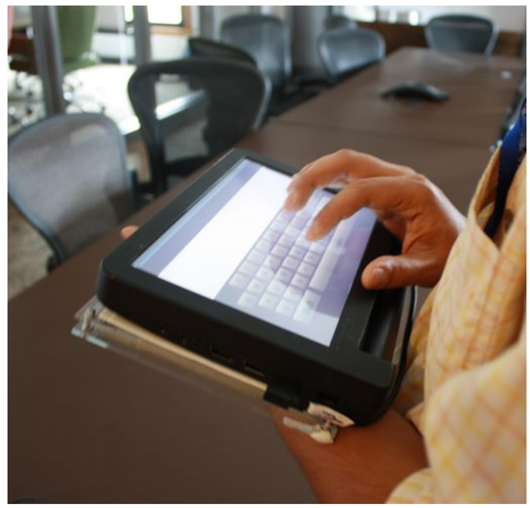

Figure 1. A user holding a foot-scale device and trying to enter text. It can be seen how this is unstable and unnatural.

Moreover, the tablet form-factor introduces two problems for the user. First, since a tablet is very portable, it is an ideal device for use while mobile (e.g. actually moving, rather than nomadic, meaning in different places, but mostly stationary). Doctors in a busy hospital, commuters standing on a train and TV producers on location at a shoot are all examples of users who would benefit from using a computing device while standing, walking or sitting and briefly holding their device.

Unlike a mobile phone, however, a larger (e.g. 7+ inches wide) keyboard (soft or otherwise) does not work well with two thumbs because of the relatively large distance that the thumbs must travel to reach the appropriate keys, and the weight of the device compounds the problem of typing by straining the hands to support the device while making precise movements to position the thumbs over the proper keys. (drawing) While standing, many users resort to a different pose, in which one arm supports the device while the other hand types with a single finger.

A second difficulty, common to all soft keyboards, arises because the fingers do not receive any tactile feedback as to their position on the keyboard. Unlike a physical keyboard in which the keys have ridges that delineate their boundaries, a soft keyboard often requires the user to look at the keyboard to ensure that they are activating the correct keys. Looking at

Submitted for review to CHI 2011.

Figure 2. A user holding a foot-scale device in a natural and stable posture. The user is able to see his finger positions on the screen on the front.

the keys while typing is distracting, requires many eye shifts (since the user must also verify on the screen that what they are typing is correct). This can be alleviated to some extent by providing haptic feedback when a key is activated, but this provides only limited improvement.

In this paper we present novel methods that investigate the use of a backside touch input device for text input on a foot-scale mobile device. We implement two mechanisms, one of them being a backside QWERTY keyboard, the other one being a chording mechanism. We compare and contrast the performance of these novel mechanisms with a standard soft QWERTY keyboard.

RELATED WORK

Back side trackpads or touch screens (e.g. a secondary touch input on the back of a device that augments the input from the front) exist on some recent devices, including the Notion Inks Adam tablet (http://notionink.wordpress.com/), and Samsung has applied for a patent for a method to use certain multi-touch gestures on the back of a device in collaboration with a front multi-touch panel for inputing gestures (US 20100188353).

Some recent efforts have also investigated alternate methods of text inputs. Some of them like BlindType (http://www.blindtype.com) try to let the user type without doing visual search, and by doing ambiguity resolution and error correction. Some other efforts like SWYPE (http://www.swypeinc.com) investigate approaches more disconnected from traditional keypads. SWYPE is a mechanism that allows users to draw spellings on the keypad, instead of pressing individual keys.

Wearable computing users often use one-handed chording keyboards as their text input mechanism. For example the Ekatetra (http://www.ekatetra.com/), or the Twiddler (http://www.handykey

One handed chording keyboards are almost as old as computers themselves, with Doug Englebart demonstrating one in 1968 after many years of work (http://sloan.stanford.edu/mousesite/1968

Chording can be used to free up a hand for, e.g. supporting a device. Unfortunately, this requires a secondary device for the user, if a standard chording keyboard is to be used. Peripherals are problematic because they get lost, are often stuffed away in a bag, an generally inconvenient for spontaneous use.

With training, chording keyboards, in conjunction with phonetic encoding of words can be used to dramatically improve the speed of text entry. Stenotype keyboards, used by court transcriptionists and closed captioners, allow for transciption speeds in excess of 300 words per minute for some (well trained) users.

SCENARIOS

We envisioned some scenarios where people might need alternative text input mechanisms. For each scenario, we then proposed a possible mechanism that could be helpful in entering text with accuracy, and reasonable speed. It should be noted that the theme of research was not to necessarily outrun soft QWERTY in terms of speed, but to propose mechanisms that could work with reasonable performance in scenarios listed.

Stationary and visually focused

This is the best case scenario, and also the one that soft-QWERTY keyboard is perfectly suited for. In this scenario, the user is stationary (not walking or moving) and can adjust to reach an optimal configuration. This includes resting the device on the lap, and entering text similar to an actual keyboard.

Mobile and visually focused

This is the case where the user is moving but has enough time to do operations that require visual focus on the screen. This includes scenarios like commuting (standing or sitting). The challenge here is that due to occasional loss of focus while entering the text, users have to orient themselves again after each lapse. Such scenarios would require mechanisms that don't necessarily require the used to re-orient themselves time and again. In short, mechanisms designed for such scenarios would allow the user to constantly touch the screen and still be able to signal input, as and when required.

Mobile and intermittently focused

This is the worst case scenario, and it occurs in cases when the user is otherwise involved in some activity, but still has a need to enter text. This includes walking, when the user is actually focused on the path and is entering text on the side. Being able to enter text in this scenario, on devices of larger form factor is a challenge. Such scenarios would require mechanisms that have unique formations or representations that the user can memorize over time. This also means that the mechanisms would reduce the amount of visual search that the user has to perform, in order to enter a particular character.

SYSTEM

Hardware implementation

We created a hardware prototype using a Stantum Slate PC and a Stantum multitouch panel[Figure]. The multitouch panel was fixed on the back of the Slate PC. This replicated the same setup as a foot-scale device with a backside touch input, and enabled us to test out the mechanisms we had implemented.

Software architecture

One of the major functionalities of the interfaces was projecting the touch points on the backside multitouch panel, onto Slate PC's screen. To this effect, the software architecture for all the mechanisms was split into two modules; the eventlogger and the GUI:

Eventlogger

The eventlogger was responsible for capturing the events being generated by the backside multitouch panel. These events were then redirected to the GUI using a local socket connection. Since the volume of the events being generated was huge, we had to make modifications to the message passing routine of the eventlogger. Instead of redirecting all the events generated on the panel, we maintained a list of cursors or touch points and updated the cursors every 200 milliseconds. Whenever there was a change in the position or state of a cursor, we would send an update on the socket. This made sure that the socket is not flooded with information.

GUI

The UI for the mechanisms was created in actionscript 3.0. The design for the individual mechanisms will be explained in a following section, but on a higher level the GUI received cursor updates on the local socket. Depending on the type of update, cursor positions were updated, expired cursors were destroyed and new cursors were introduced, as and when required.

EXPERIMENT

Participants

For the purpose of the study we tried to keep the backgrounds of the participants mixed, but also ensured that all of them have decent exposure to typing on QWERTY keyboards (not necessarily soft-QWERTY). The participants in the study were all working professionals with more than 3 years of experience (on an average) with QWERTY keyboards. According to post-test interviews, out of the total of 36 participants in the study, 32 participants had prior experience with soft-QWERTY keyboards. This information was important as familiarity with a particular style of text-input mechanism gets reflected in the kind of speed people achieve with the same. Though we tried to achieve a reasonable mix of backgrounds for participants, we acknowledge the fact that we still had a sample that was experienced. Therefore, our results are representative of participants who have had reasonable level of exposure to typing on QWERTY keyboards. For novices, the results might be different, in any direction.

Phase 1: Usability test

After we implemented the QWERTY, backside QWERTY and chording mechanism, we conducted two rounds of usability evaluation. The first one with 3 participants, and the second one with 6 participants. For the purpose of the usability study we picked a text corpus that was different from the corpus used in the phase 2. Both the corpora were generated by choosing randomly from Scott Mackenzie's text corpus [Reference], and were mutually exclusive. The statistics for the two are listed in Table 1. The participants in the usability test spent 20 minutes familiarizing themselves with the interface and were then asked to enter the entire usability test corpus. The entire process was captured on videos and post-test interviews were also conducted. Based on the post-test interviews and analysis of videos, changes to the interfaces were made.

Phase 2: Scientific experiment

Conditions

The experiment had three conditions. As mentioned earlier, there were a total of 36 participants. Each condition had 12 participants. The three conditions were:

- QWERTY
- Backside QWERTY
- Chording Mechanism

Data collection methods

To make sure that all the important aspects of the experiment and the feedback from the users is captured fully, we used multiple data collection methods:

- Videos: All the sessions were fully recorded. In total, around 20 hours of videos were recorded, by the end of the experiment.
- Data Logs: All the mechanisms had a built in data logging feature, that recorded each and every action of the user, along with timestamps. This helped in understanding parts of the videos, where the user was stuck or had trouble accomplishing what they wanted.
- Post-session interviews: After each session, the participants were asked to report on their experience with the interface. To give the discussion some structure, the users were asked the following questions:
    1. What did you like the most?
    2. What did you dislike the most?
    3. What would you change in the interface?

	Experiment corpus	Usability test corpus
Average phrase length	15.07	15.67
Number of words	76	81
Unique words	62	67
Min. length of word	1	1
Max. length of word	11	12
Average word length	4.95	4.43
Number of characters	437	425
Correlation with English	0.9297	0.9377

Table 1. Statistics for text corpora

- NASA task load index: To be able to quantitatively capture the experience with the interfaces, the NASA task load index was used. The NASA Task Load Index (NASA-TLX) is a subjective, multidimensional assessment tool that rates perceived workload on six different subscales: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. It was developed by the Human Performance Group at NASA's Ames Research Center over a three year development cycle that included more than 40 laboratory simulations [Reference] [Reference]. It has been cited in over 550 studies[Reference] and a recent search for NASA-TLX on Google Scholar revealed over 3,660 articles. These statistics highlight the large influence the NASA-TLX has had in Human Factors research. Therefore, we chose it as a tool in our experiment to capture the user experience.

Measures

- Keystrokes Per Character (KSPC)
- Words Per Minute (WPM)
- Speed vs Accuracy tradeoff

Process

The participants in the study were supposed to go through the following steps during the study. Care was taken that the steps remain the same across all participants so as to control the environment. Ideally, the mechanisms should have been tested out in the scenarios that we had earlier listed. However, the lack of prior research on the topic, suggested that the first few research cycles should be conducted under controlled conditions. This eliminated the possibility of the environment acting as a confounding variable in the experiment.

1. Participants were briefed about the goal of the session. They were also briefed about the structure of the session.
2. They were given a brief introduction to the input mechanism. This was done by one of the researchers.
3. They were given a the test corpus and asked to spend the next 20 mins familiarizing themselves with the input mechanism. The text they were supposed was the test corpus.
4. The entire process was videotaped for data analysis and validation purposes.
5. After 20 minutes, they were handed over the experiment corpus. They were asked to input the corpus in its entirety, using the mechanism that they had just encountered. They were asked to be accurate with their input, and the system would underline their mistakes as and when they occur.
6. Once the participants had entered the entire text without any errors, they were handed the NASA TLX questionaire and asked to rate their experience. Since the index is relative they were asked to compare their experience with their previous exposure to a soft QWERTY keyboard. This was also done for the QWERTY mechanism, just to make sure that the mechanism is a fair representation of the soft-QWERTY family. Since the NASA-TLX is a 20 point scale, we asked them to assume that QWERTY was at 10 on each scale. This was done so that the individual ratings could be studied in details during qualitative analysis.
7. Finally, they were interviewed on any other qualitative feedback they had on how to make the mechanisms better.

MECHANISMS

As mentioned earlier, we implemented two backside touch input based mechanisms and a standard soft-QWERTY keyboard. The details of each are as following:

QWERTY

This was a standard soft QWERTY keyboard [Figure], with no special modifications. The keyboard supported multi-touch, which means the users could select the next character to be entered without releasing the currently selected one. The keys turned blue on click, so that the user gets appropriate feedback. The top quarter of the screen was a scrollable textfield, which displayed the text that was being entered.

Backside QWERTY

Description

This mechanism used the standard QWERTY layout, but with a few modifications. In this mechanism, the user could place his fingers on the backside screen and it would result in a cursor on the front screen at a location that is vertically above the touch point. This way users could move around multiple cursors at the same time, using multiple fingers. To input a particular character, the user was required to go select a particular key with the cursor and then touch the front screen anywhere to signal input.

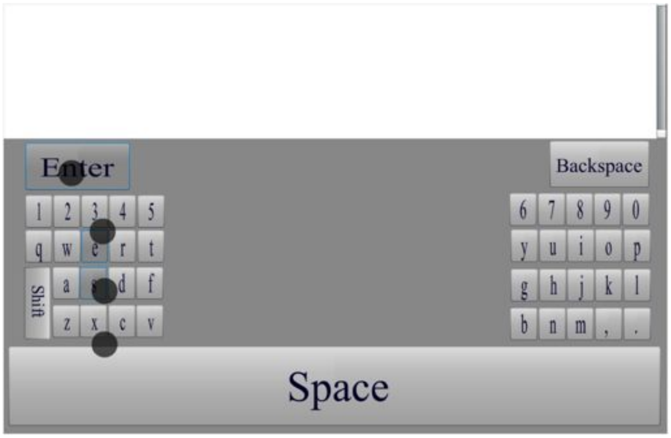

Figure 3. Screenshot of backside-QWERTY with user's fingers on the backside screen

Design evolution

The interface went through two iterations before getting its final shape. The first version used pressure as a method to signal input. However, usability tests suggested that being able to control the amount of pressure being applied is hard for users. Since the users were already applying some pressure to drag the cursors around, it turned out to be confusing for them.

Therefore, the second version of the interface used cursor destroy events as a method to signal text input. In simple words, the interface would accept a character as input if the corresponding key had a cursor destroy event over it. However, in this case the usability tests suggested that choosing not to give an input would mean that the user needs to go to section of the screen that has no keys and then choose to remove the touch/cursor. Also, the chances of giving accidental inputs, by accidental destroy events increased menifold. Both these factors resulted in additional overhead in terms of time.

As a result we finally shifted to touching the front panel once the character has been selected, as the method to signal input.

In the first version we had also modified the keyboard layout to match the finger to key mappings on a QWERTY keyboard [Figure]. However, after the usability test, we realized that users were still doing visual search for keys instead of using their pre-existing knowledge of QWERTY. Therefore, a QWERTY layout in this case turned out to be more predictable and usable.Moreover, the size of the Space, Backspace and Enter keys was increased so that they are easy to select.

Chording mechanism

Description

In this mechanism the user was required to use the number of fingers and the x-position of the fingers to signal the character that was being entered. The screen in this case was split into 6 zones (3 on left, 3 on right) and each zone had 3 segments. Users could switch from one zone to the other by moving a cursor into that zone. Just like in backside QWERTY, the users were still touching the backside screen in order to move cursors on the front. The segments had levels corresponding to them. The number of fingers in a particular zone determined which level of segment the user has selected. If the user wanted to select the first level segment in a particular zone, they would have to move one cursor to the zone. If they wanted to select the second level segment, they were required to move two cursors to the corresponding zone, and so forth. Each segment on the left side of the screen corresponded to 9 characters. The characters were organized in an alphabetical order [Figure], so that visual search is simplified. These 9 characters would open up on the right side if the user selects the corresponding segment on the left. To select a particular character that was over a segment, the user was supposed to select the segment (as explained above) and touch anywhere on the front screen. Therefore, each character corresponded to a unique "chord" on each side of the screen, and therefore the mechanism was called the chording mechanism.

The use of number of fingers, as opposed to position to select a segment was done deliberately to restrict the movement in one dimension. This was done in order to make sure that given enough time the user is able to remove his eyes off the interface while entering text. Moreover, error rate was expected to go down, because absolute positioning was restricted to just one dimension.

Similar to backside QWERTY and QWERTY, the top quarter of the screen was a scrollable textfield, which displayed the text that was being entered.

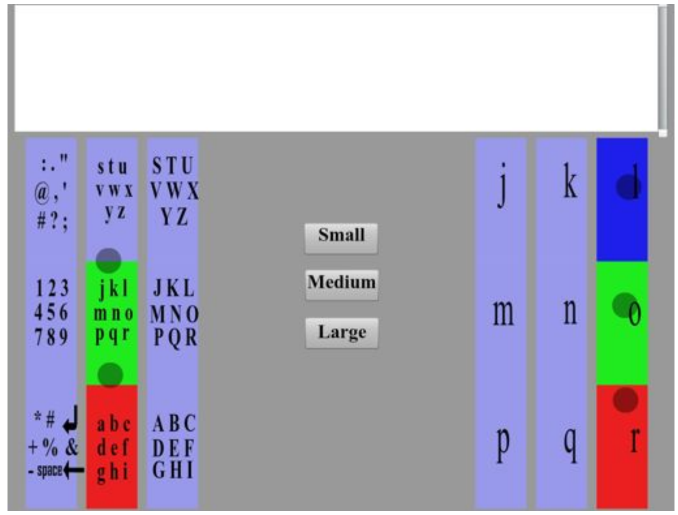

Figure 4. Screenshot of chording mechanism with user's fingers on the backside screen trying to form a chord

Design Evolution

As in the case of backside QWERTY, after the first round of usability testing, it was observed that users find it hard to control the pressure being applied by the fingers. Therefore, in the final version of the mechanism the users could touch the front screen to signal text input for the character selected by the chords. Usability testing suggested that the finger sizes and areas in which users can move comfortably vary a lot. Therefore, in the final version the users were allowed to select from large/medium/small setting [Figure], that determined the width of the zones. Users, who had smaller reach, could use these settings to optimize the area of movement.

RESULTS

Quantitative results

We will discuss our quantitative results in terms of the three measures that we have proposed earlier in the paper.

Keystrokes Per Character (KSPC)

KSPC is generally treated as a measure of accuracy, because it represents the number of keystrokes executed per character. From our experiment, it turned out that all the three mechanisms were similar in terms of KSPC measurements as shown in Table 2. The Chording mechanism was on an average faster than the other two mechanisms, with the Backside QWERTY being the slowest in terms of average. However, when we did a t-test between KSPC measurements for QWERTY and chording mechanism, the p-value turned out to be 0.35. This suggests that over the test subjects there was no significant difference between the accuracies of the QWERTY and the chording mechanisms. Similarly, a t-test between KSPC measurements for QWERTY and backside QWERTY resulted in a p-value of 0.24, which still lacks significance. Therefore, both the backside touch input mechanisms were as accurate as the QWERTY mechanism.

Words Per Minute (WPM)

Words Per Minute (WPM) is a measure that is commonly used to represent the speed of a text input mechanism. Our experiment findings suggested that QWERTY was still the fastest mechanism, followed by backside QWERTY and the slowest was chording. Statistics on WPM measurements can be found in Table 3. The results suggested that QWERTY was faster than both backside QWERTY and chording mechanisms. However, it should be noted that the test sessions were generally around 20-30 minutes, and each user only interacted with a mechanism once. Therefore, the fact that backside QWERTY was on a average 3/4th as fast as the QWERTY mechanism was encouraging. This led us to explore the results qualitatively and also in terms of speed versus accuracy trade-off.

Speed versus Accuracy Trade-off

In spite of the discussions above, we should acknowledge that none of these measures can be studied totally independently, and there was a possibility that the participants were being more accurate by sacrificing on speed. However, since speed with a particular input mechanism is often attributed to the amount of exposure and practice, we had reasons to believe that the comparison of accuracies was still fair. However, we still did some Speed vs Accuracy analysis for the three mechanisms and [Figure] is a plot of the same.

Sample tests

Since the amount of exposure that the users received during the sessions was limited, it was obvious that lack of experience with the mechanisms is also hampering the speed and accuracy measurements. Therefore, one of the researchers who was involved in development of the interface and had reasonable exposure to the interface went through the test in exactly the same fashion as the participants. This was done to test the capability of the two new mechanisms in terms of speed and accuracy. Table 4 shows the measurements from the same. It can be seen from the table that with decent amount of exposure to the interface, both the accuracy and the speed seem to show better trends. However, these measurements are restricted to an individual and are highly preliminary. To fully establish our claims, larger and longer studies would have to be conducted.

Qualitative results

As mentioned earlier, the NASA task load index was used to get subjective ratings on qualitative aspects of the interfaces. This was just done for the two new mechanisms that were implemented. This was done deliberately because the participants were asked to give the ratings keeping in mind their experience with soft-QWERTY keyboards. Since all the participants in the study had prior exposure to soft-QWERTY keyboards, this factor was uniform through the experiment. In the following few paragraphs we summarize the high-level trends that were derived from the ratings that the participants assigned to the backside QWERTY and chording mechanisms. Table [5] and [6] present the ratings given by the users to the two mechanisms.

Mental Demand

By just looking at the ratings on the mental demand of the task on the NASA-TLX scale, it seemed that the users reported less mental load for the backside-QWERTY as compared to the chording mechanism. When we conducted a t-test on the ratings, it turned out to have a p-value of 0.02, which suggests that the difference was significant. It also means that the backside-QWERTY has significantly less mental load than the chording mechanism. However, the average for backside-QWERTY was 29.5 and that for chording was 43, which means that on an absolute scale participants thought that the two mechanisms were not mentally intensive to work with. The chording mechanism was deemed as harder to understand because in that case, the users were trying to work with the number of fingers as a method of input. This setup was new for all the participants in the study, and this also relates back to the low speed (in WPM) of text entry on the chording mechanism. The averages are lower than 50 for both the mechanisms, which denotes that quite a few participants believed that the mechanisms are simpler to understand and use than QWERTY. This effect was more well defined for backside-QWERTY (average of 29.5), as opposed to chording (average of 43).

Physical Demand

An analysis of the physical demand ratings on the NASA-TLX suggested that the participants in general found the chording mechanism to be as physically challenging than the backside-QWERTY. Looking at the kernel density plots of the two revealed that the distribution of population across ratings was very similar in the two interfaces. Also a t-test on the two sets of ratings resulted in a value of 0.91, which

	Avg. KSPC	Max. KSPC	Min. KSPC
QWERTY	1.288	1.325	1.239
Chording	1.261	1.373	1.110
Bksd. QWERTY	1.338	1.497	1.278

Table 2. KSPC Statistics

	Avg. WPM	Max. WPM	Min. WPM
QWERTY	12.162	15.233	10.2
Chording	4.061	4.79	2.722
Bksd. QWERTY	8.598	10.055	5.432

Table 3. WPM Statistics

	WPM	KSPC
Chording	7.69	1.12
Backside QWERTY	13.231	1.152

Table 4. Sample Measurements

meant that the difference was not significant. Therefore, it is reasonably fair to say that the two mechanisms were equally easy or equally hard to use. However, since the averages of both the sets of ratings were around 50 (10 on the original scale) it means that none of the mechanisms were exceptionally hard to use, as opposed to each other. This also means that the two mechanisms were on an average as easy to use as the QWERTY mechanism, since the participants were assuming the QWERTY mechanism to have a rating of 10 (on the 20 point scale) for all metrics.

Temporal Demand
This metric was important in the sense that we wanted to make sure that the participants do not feel rushed during the task. Our aim was to reproduce the natural experience of entering text, as far as possible. Therefore, the low average scores (27.5 for both mechanisms) suggested that the task was not pushing the participants to an extent that they start noticing it. There were constraints that we had specified, but none of them seemed to upset the participants. The fact that there was no time limit to the task, was helpful in this respect.

Performance
The NASA-TLX index ratings for performance suggested that both the interfaces had good performance. The average performance ratings for both the interfaces were below 8 (on 20 point scale). It should be noted that a low rating on this scale means good performance. We also observed that there were two users who gave the chording mechanism a higher rating, thereby implying that it did not have good performance. Both these participants had writing speeds that were lass than the average. This suggests that these participants were struggling to get accustomed to the device and the mechanism. Their speed was suffering as a result of the same. We also looked at the videos from those sessions, and they corroborated the same claim.

Effort
The difference between the sets of ratings for backside-QWERTY and chording was not significant. The t-test results in a p-value of 0.52. However, studying the kernel density plots [Figure] more carefully suggested that the distribution of the opinion on the amount of effort involved in working with the backside-QWERTY was bimodal. However, for the chording it was almost evenly distributed around the average rating. The videos suggested that some of such cases were because the backside-QWERTY did not allow for change in size of the keyboard, people who had fingers longer or shorter than average finger sizes had a harder time with the mechanism as opposed to other. The chording mechanism on the other hand, did allow for such changes and therefore got an even distribution of ratings.

Frustration
The ratings suggested that on an average users were satisfied with the mechanisms. The frustration levels/ratings were restricted to the lower half of the scale for chording, however only two users reported higher levels of frustration with the backside-QWERTY. When we cross-checked this with our video logs, it turned out that both of these users had issues with getting accustomed to the mechanisms primarily because of finger sizes, as pointed out in the last section. This in turn resulted in the observed frustration on the NASA-TLX.

Mean Weighted Scores
After the individual analysis of the ratings, we used the standardized methods to calculate the effective weighted task load ratings, as specified by NASA. Right after filling up the survey, the participants were also asked to compare metrics against each other. Since there were 6 metrics (Mental, Physical, Temporal, Performance, Effort, Frustration), there were C_2^6 possible combinations, and 15 questions in total. After receiving all the responses from the participants, we found that on an average effective weights for Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, Frustration were 4, 3, 3, 1, 2, 2 respectively. In simple words, a higher weight means that particular dimension or metric has larger effect towards the load of the task and should be given more weightage as opposed to others. Initially we thought that interactions with backside-QWERTY and chording mechanisms were intrinsically different tasks, and therefore the weights should not be treated as the same. However, the average weights that were calculated for the two mechanisms turned out to be very close to each other. This is understandable from a viewpoint that both the tasks were actually text entry tasks with same corpus, and same kind of input method. Therefore, the value that the participants were attaching to each metric didn't change. It is apparent from the discussion above, but to clarify again, a low weighted score on the NASA-TLX means that the overall task load was low and the experience was pleasant. The backside-QWERTY obtained a weighted score of 37

	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration	Mean Weighted Score
User 1	15	70	15	15	30	25	29
User 2	15	40	20	20	30	24	24
User 3	20	75	25	25	30	10	32
User 4	25	65	25	10	35	25	33
User 5	20	60	35	35	45	35	37
User 6	25	30	30	20	30	40	29
User 7	15	75	30	55	65	15	39
User 8	45	35	40	30	40	35	39
User 9	55	80	20	75	50	75	56
User 10	60	40	35	25	40	85	49
Avg. Scores	29.5	57	27.5	31	38.5	37.5	37

Table 5. NASA-TLX rating for backside-QWERTY mechanism

	Mental Demand	Physical Demand	Temporal Demand	Performance	Effort	Frustration	Mean Weighted Score
User 1	20	20	15	20	30	15	20
User 2	25	35	25	25	25	20	26
User 3	25	55	35	30	45	40	38
User 4	30	60	40	35	55	45	44
User 5	35	70	30	10	50	35	41
User 6	15	75	10	70	35	30	34
User 7	65	80	30	75	40	25	53
User 8	70	85	45	15	45	40	57
User 9	80	60	25	30	45	40	52
User 10	65	40	20	55	35	35	42
Avg. Scores	43	58	27.5	36.5	40.5	32.5	41

Table 6. NASA-TLX rating for chording mechanism

on an average, and the chording mechanism got an average score of 41. The t-test between the two resulted in a p-value value of 0.22, which means that the difference between the two sets of effective ratings was not significant. The weighted scores for both the mechanisms were low, and since the participants were comparing the mechanisms with their experiences with soft-QWERTY, it can be seen that the two new mechanisms were definitely welcome by the participants. From a qualitative standpoint, the mechanisms seemed to create good user experience, even better than a soft-QWERTY in some cases.

Design guidelines

After a qualitative and quantitative analysis of our results, we also did a high-level analysis of our design choices and cross-checked them against the videos. As a result of this, we came up with some major design takeaways from this piece of research. Therefore, in this section we propose some design guidelines for future efforts that look into text input by utilizing a backside touch input device. These guidelines are not sufficient, but should definitely be treated as necessary.

Movement minimization

During the study we realized that the amount of movement involved in selecting a particular character determines the speed that users would achieve with the mechanism. The post-experiment analysis of usability test videos corroborated this claim. Once we reduced the size of the keys and magnified the movement of fingers, the users could cover larger distances with smaller shifts in position. We also realized that users are able to control the finger position with very high accuracy, and therefore these optimizations help them enter text at higher speeds.

Multiple finger sizes

There can be a lot of variation in finger sizes, amongst users. We accounted for this in the chording mechanism, by having settings that user could select, if they had fingers larger or shorter than the average. This was critical for chording mechanism as the users were trying to form chords at specific locations. For backside-QWERTY, we did not make this optimization because users were not trying to position multiple fingers at the same time, and also because in that case we had tried to optimize between finger movement and key sizes. Dynamically determining the trade-off between the two, depending on the finger size would have interfered with the optimal setting of the system, and influenced other factors.

Reducing dimensions of movement

This one is only true for the chording mechanism, but it turned out from the experiment that a good way to maximize on accuracy is to reduce the number of dimensions of movement. Traditionally, with a soft-QWERTY users tend to position themselves in both, x and y co-ordinate. In the chording mechanism, the y direction was being controlled by the number of fingers, and therefore the movement was just restricted to the x direction.

Visual search vs Recall

After the usability testing, we also realized that users tend to do a visual search to find characters instead of recalling from their previous experiences with QWERTY mechanisms. It turned out the the layour of keys should be visually intuitive and familiar. As long as the positions of characters follow a pattern, either pre-existing (like QWERTY) or familiar (alphabetic), users will be able to accustom themselves in a few interactions.

Pressure vs Touch

As explained earlier, we also experimented with using pressure as a mode of input, but it turned out that it is hard for users to accurately control the amount of pressure being applied. This results in high error rates because of spurious inputs. A design fix that we used to mitigate the situation was to use touch on the front screen. Since the two thumbs are anyway used to hold the device, it was easy for the users to use them to tap on the front screen. This significantly reduced error rates from the test version to the final version.

Touch Cursors

Both our mechanisms involved showing finger positions on the screen, and we had to this in a way that we don't hide any information or don't cause a loss of perception. We achieved this by doing a number of things. We made the touch cursors transluscent, so that we don't occlude any information. We also kept the size of the cursor smaller than the size of an individual key so that they are easier to position and don't end up selecting multiple keys at the same time.

FUTURE WORK

This piece of research was one of the few recent efforts in the domain of text-input mechanisms with backside touch input. There are many directions that can serve as follow up to this work. However, the future work can largely be organized into categories as follows:

Haptic Feedback

In the next phase, we would want to include haptics and tactile feedback. This would entail producing vibrations and other cues to signal a change on the interface. Especially in the case of the chording mechanism, whenever the user switches from one zone to the other, or selects a new segment, it should be coupled with some feedback. This would also help the interface to be used by visually challenged people as well.

Proximity Sensing

As of now, the users just see the position of the fingers on the backside touch. Though the participants in our study were able to work well with this setup, we feel that for wider acceptance the interface should be able to track fingers with more robustness and greater detail. We plan to use proximity sensors that would determine how far the fingers are from the screen, and try to estimate the position of the cursors even without touching the screen. This would allow for faster positioning and orientation. We hope that this change would help the users enter text even faster.

Longitudinal deployments

The time that the users spent on the mechanisms was limited and short. They had to get accustomed to the interface in 20 minutes and then start on the test. In the future, for stronger results we would want to run longitudinal deployments. This would give users enough time to get used to the interface, and then our measurements would be comparable to the more frequently occurring cases like QWERTY.

Modified Chording

To further minimize the amount of movement in the chording mechanism, the future iterations of the mechanism would use the first touch to select the zone, and the rest of the touched would select segments irrespective of position. Therefore, the user won't be required to try and bring 1-3 fingers into a particular zone. Instead they would just take one finger to a particular zone and then place the others at any random location thereby selecting the segments in the zone that was selected by the first finger. This would considerably reduce the effort involved in the chording mechanism, and would also make the interface even more ready for "blind" text entry.

Testing in valid scenarios

An earlier section talked about how there were some scenarios that we envisioned and observed. The concepts of the text-input mechanisms were based on these scenarios. However, since this was the first evaluation of this sort we wanted to control as many variables as possible. Therefore, very similar conditions were reproduced for all the participants. In the near future we would want to test the applications in the scenarios in which they were supposed to serve and then evaluate the same, both qualitatively and quantitatively.

REFERENCES