Improving text entry performance on tablet devices

Élvio Rodrigues Micael Carreira Daniel Gonçalves
Department of Computer Science and Engineering
Av. Rovisco Pais, 1000 Lisboa
{elvio.rodrigues,micaelcarreira}@ist.utl.pt, daniel.goncalves@inesc-id.pt

Abstract
In order to improve the text entry speed and error rate on tablet devices, we developed and tested 5 virtual keyboard variants. Some variants try to avoid errors by highlighting the next four most probable keys, either by changing its width or its color. Other variants were designed to decrease neighbor substitution errors, by shifting users' taps or by increasing the underlying area of the keys, based on its probability. The developed keyboards were tested by twenty young adults. Results show that soft keyboards without visual changes are the fastest method for text entry. Also, the use of word prediction further decreases typing speed, without improving the error rate. The Shifted and Size Invisible variants reduced neighbor substitution errors by 48.65% and 62.96%, respectively. Further improvements on error rate remain possible if we combine the strengths of multiple variants into one single variant.

Keywords
Text-entry performance, multi-touch tablets, typing speed, error types, pre-attentive interfaces

1. INTRODUCTION

Touch devices are being increasingly used for a wide variety of tasks. However, these new and updated technologies lack the haptic feedback of physical buttons, making it harder to accurately select targets. This characteristic hampers certain tasks, such as text-entry, where the user has to constantly select one of many small targets. This is also the reason why text entry on touch devices remains slower and more error-prone than on traditional computer keyboards.

While some studies report that touch events are slightly skewed towards the bottom-right of the screen in smartphones [Henze12, Henze11], the veracity of such patterns remains to be proved for tablets. Therefore, throughout this paper, we analyze the text entry data we collected from 20 young participants using a traditional virtual QWERTY keyboard and five variants. These variants use letter prediction to create pre-attentive interfaces, word prediction, shifts touch events and increases the underlying area of the most likely keys. Then, we systematically analyze the performance of each variant, and report the traditional text-entry measures of words-per-minute (WPM) and error rates. We also discuss in more detail the different kind of errors that users do – insertion, omission and substitution/cognitive errors.

We conclude that a virtual QWERTY keyboard without visual changes is the fastest method for text entry. Also, the use of word prediction further decreases typing speed, without improving the error rate. However, when the most probable letters are highlighted with a brighter color, a significant error decrease is verified. Some variants are able to correct most of neighbor substitution errors, but these improvements are not significant in the overall error rate. Furthermore, our results did not confirm the tendency of tapping on the lower right corner of targets.

2. RELATED WORK

Although there is a relative extensive body of work regarding text entry on touchscreen devices focusing more on smaller-sized devices such as smartphones, not much research has been done regarding tablets. Still, some of the categories on which we will focus are orthogonal to that fact.

2.1 Shifted Touch Events

It is widely known that when users try to acquire a target on a touchscreen, they actually touch on the surface with systematic error offsets [Holz10]. Henze et al [Henze11] analyzed the touch behavior of smartphone users through a game they published into the Android Market. After analyzing the data, the authors concluded that events are systematically skewed towards a position in the lower-right screen.

These offset errors were later verified by various authors for text input also. For instance, Henze et al. [Henze12] developed a typing game that recorded how users touch on the standard Android keyboard to investigate their typing behavior. Results show that users' taps are systematically skewed towards the bottom of the screen along the vertical axis.

2.2 Adaptive Keyboards

In order to solve the offset error, Himberg et al. [Himberg03] developed a method for on-line adaptation of a touch pad numerical keyboard layout. The algorithm subtly moves the keys according to the spatial distribution of keystrokes. In consequence, the keyboard matches better the users' physical extensions and grasp of the device, and makes the physical trajectories more comfortable during typing.

Findlater et al. [Findlater12] also evaluated two novel personalized keyboard interfaces specifically for ten-finger typing, both of which adapt their underlying key-press classification models. One of the keyboards also visually adapts the location of keys, while the second always maintains a visually stable rectangular layout. Results show that the *NonVisual-Adaptive* keyboard provided a typing speed improvement over *Conventional* (baseline keyboard), but *Visual-Adaptive* did not (visualizing adapted key layouts can negatively impact speed).

2.3 Personalization

As noted by Cheng et al. [Cheng13] on a recent study, people use different hand postures to type on tablets, depending on the situation. This study showed that 98% of the users preferred different keyboard layouts and positions depending on how they were holding these devices. The authors developed *iGrasp*, which automatically adapts the layout and position of virtual keyboards based on how and where users are grasping the devices without requiring explicit user input.

Since different hand postures leads to different touch typing patterns, Yin et al. [Yin13] highlighted the importance of taking this information into account when developing a personalized adaptive keyboard. Therefore they proposed a new approach for improving text entry accuracy on touchscreen keyboards by adapting the underlying spatial model to factors such as input hand postures, individuals, and target key positions. A specific sub-model is only applied if its corresponding input posture can be identified with confidence, and if the sub-model has enough training data from the user. The authors report that when posture, user, and key adaptations are combined, they achieve the greatest improvement.

2.4 Language Models

Another way to significantly reduce the error rate of soft keyboard usage is through language models combined with models of pen placement, as emphasized by Goodman et al [Goodman02]. When a user hits a key near the boundary of a key position, both language model and key press model can be used to select the most probable key sequence, rather than the sequence dictated by strict key boundaries. Results show that this can lead to an overall error rate reduction by a factor of 1.67 to 1.87.

Several approaches to highlight keys have been studied which involve making the rendered keys larger or smaller, depending on their likelihood [AlFaraj09], or labeling the corresponding keys in bold [Magnien04]. Still, some studies [Himberg03] report that users could find the dynamic rendering of keys distracting. In order to avoid the aforementioned distraction, Gunawardana et al. [Gunawardana10] developed a method that expands or contracts the keys' underlying area, based on a language model. A simulation suggests that it reduces the errors rate. Finally, several researchers have developed alternative keyboard layouts based on Fitt's law and character level bigrams such as the Metropolis [Zhai00] and OPTI [MacKenzie99] layouts.

3. TEXT PREDICTION

In order to develop more advanced variants of the virtual QWERTY keyboard, we used two types of prediction to anticipate what the user is going to write: *word prediction* and *next letter prediction*. If the *prediction system* is able to guess correctly, the number of keystrokes needed to write a sentence decreases. This way, it can also enhance the speed of writing and reduce the physical effort required to compose messages. In addition, the prediction software may also fix spelling mistakes, reorder sentences and more generally enhance the quality of the composed messages. The most advanced prediction systems have learning features, are able to make inferences, are adaptable and are able to act independently [Garay-Vitoria06].

There are several techniques to predict the text the user is trying to input, some more complex than others. However, by increasing the complexity of the predictions systems, the prediction results only increase marginally [Garay-Vitoria06]. This way, and since the aim of this study was not developing a novel and more efficient prediction algorithm, we opted for a simplistic one. Our predictor only takes word frequencies into account and, when the user writes the beginning of a word, the system offers the most probable words beginning with the same character(s).

To implement the *word prediction system*, we used the CETEMPúblico Portuguese text corpus[1], which contains approximately 180 million words. From that corpus we processed the word frequencies and then stored it in a dictionary structure that contains all the information about each word and its prefixes frequencies, so that the information can be efficiently accessed. When the user is typing, the predictor shows an ordered list of the most frequent words that start with the typed prefix.

After implementing the *word prediction system*, we decided that the *next letter prediction* should be based on the same algorithm in order to avoid the case of the *letter prediction algorithm* suggesting a letter that is not present in any of the suggested words. For instance, imagine the user wants to write "home", and at this point has already typed "ho". If the *letter prediction algorithm* suggests the letter "t" (hot) and the *word prediction system* suggests the word "home" it could be confusing for users. So we decided to implement the *letter prediction algorithm* through the *word prediction system*. What happens is, since the most probable word is "home", and the user has already typed "ho", the *letter prediction algorithm* will choose to highlight the "m" key.

3.1 Results of the prediction

To evaluate the efficiency of the implemented prediction system, we used 88 sentences that were extracted from a written language corpus from another study [Nicolau13]. Each sentence had 5 words with an average size of 4.48 characters and a minimum correlation with the language of 0.97. We then analyzed the success rate of *word prediction* by considering the frequency of the intended

[1] http://www.linguateca.pt/cetempublico

word in the list of suggested words, after writing half of the word. Only words between 6 and 12 characters long were considered, because any smaller lengths do not represent considerable savings in key presses, and above that there were not many words in our set of sentences. We performed this evaluation suggesting between 1 and 7 words.

As expected, the more the suggested words, the greater chance of success. The success rate increases almost linearly, and ranges from 30% (1 word suggested) to 81% (7 suggested words). However, the success rate does not seem to increase much when presenting a list of more than 6 words (only an increase of 3% between suggesting 6 and 7 words). We must also take into account that the more words we suggest, the more cognitive effort is required for the users to process the suggestions list. Therefore, there should be a balance between the number of words suggested (which affect directly the success rate) and cognitive effort required to process the suggestions list (which increases with the number of words).

We also performed the same evaluation for the *next letter prediction*. It is much easier to correctly predict the next letter (space included) than to predict the full word the user is typing, since the same next letter is shared for several words. The success rate increases logarithmically, ranging from 66% (1 predicted letter) to 96% (27 predicted letters). Until 4 letters, the success rate increases from 4-7% and after that, only an increase of 0-2% is found. Note that we never hit 100% success even if we highlight all the letters of the keyboard and that is because in our sentences we had a surname that was not in our prediction system, so the system could not predict it.

4. IMPLEMENTED QWERTY VARIANTS

As we stated previously, text entry on touch devices remains slower and more error-prone than on traditional computer keyboards. This way, we decided to evaluate different alternatives for the traditional virtual QWERTY keyboard, with the aim of allowing users to input text faster and with fewer errors.

Taking this into account, after developing the regular QWERTY keyboard to serve as a baseline, we developed 5 variants, which are described in the following subsections. The *Color* and *Width* variants use the *letter prediction algorithm* to highlight the next 4 most probable letters. The *Predict Words* variant is a common solution on most touch devices, which allow users to select a whole word from a list of suggested words. The *Shifted* and *Size Invisible* variants aim to reduce neighbor substitution errors, by shifting touch events and increasing the underlying area of the most likely keys, respectively.

All the keyboard variants were implemented as a Windows Metro App for Windows 8.

4.1 Color variant

The *Color* keyboard variant uses the *letter prediction algorithm* described in Section 3 to highlight the next most likely letters for the current word. Regarding the number of keys to highlight, we decided to highlight four keys because Faraj et al. [AlFaraj09] have previously tested highlighting one, two and four keys, and obtained better results with the latter. Also, the results of the *letter prediction algorithm* evaluation showed that highlighting four letters has an increased success rate when compared to highlighting fewer letters. Therefore, this is the optimum number of letters to highlight. We opted to highlight a key by changing its color from black to grey, which is a neutral color (Figure 1a). This way, the cultural connotations that are associated with particular colors are avoided (e.g.: the green and red colors have positive and negative connotations, respectively). Also, the label of the button (the letter on the button itself) increases in size. The highlight is continuous: the more probable the letter, the brighter the color and bigger the label of the button.

When we developed this variant, we thought that it would be particularly useful for users that were not completely familiarized with the QWERTY layout, because it would allow them to locate the letter they want to type faster. We also expect that users commit fewer errors by noticing if they are pressing a key that is not highlighted, or by acknowledging they missed a key press.

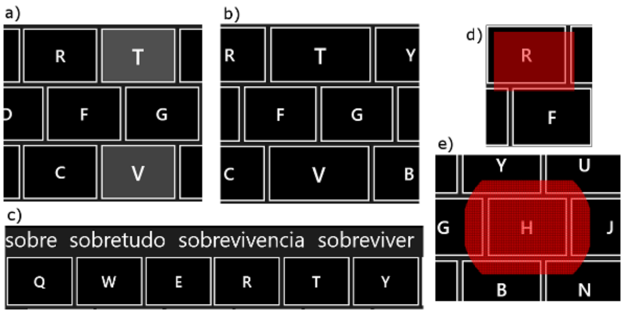

Figure 1: (a) Color variant; (b) Width variant; (c) Predicted words variant; (d) Shifted variant; (e) Size invisible variant.

4.2 Width variant

The *Width* variant uses the same principle as the *Color* variant. The difference is that it highlights the 4 most probable keys by increasing their width by 30% (Figure 1b). However, for this variant we did not use a continuous increase in size based on the probability of this letter, because it was much harder to tell which buttons were highlighted if the size increase was small. As happens with the *Color* variant, the label of the button (the letter in the button itself) increases in size proportionally to its probability.

With this variant we expect that the users commit less substitution errors by hitting the desired key instead of the neighbor keys, since the most likely keys are bigger. Also, we expect users to notice if they are pressing a letter that is not highlighted, and thus commit fewer errors. A previous study [AlFaraj09] has shown that this approach can both improve the speed and reduce errors of the typed sentences in smartphones.

4.3 Predict Words variant

This variant is a common alternative that can be selected as typing method in most of the touch devices. While the user is typing, a list of the most likely words is shown in a horizontal ribbon above the keyboard (Figure 1c). If the word the user wants to write is on the suggested list, he can save some key touches by tapping it so the full word along with a space character will be inserted.

In the literature, there is no conclusive study about the optimum number of words to suggest [Garay-Vitoria06]. Since there is a trade-off between the number of suggested words (that directly affect the success rate) and the cognitive effort required for the user to process the list, we opted to suggest 4 words.

Although this is not a novel approach, we wanted to confirm in a systematic way if this variant would possess any advantage over the normal QWERTY keyboard, either in typing speed or quality of the typed sentences (with fewer errors). It is a fact that users save some time by tapping less keys, but they also waste time in the cognitive effort of continuously checking the suggestion list.

4.4 Shifted variant

The approach of shifting the real touch area of keys from its visual representation is also common in many virtual keyboards [Henze11, Henze12]. In small touch devices, like smartphones, this approach has proven its benefits [Henze11, Henze12]. However, no systematic studies have been performed for tablet devices. These devices vary from the former not only in screen size, but also in the typing posture users assume when using them; in smartphone users usually type with the two thumbs, while in the tablet they can type with all fingers.

Previous studies have consistently shown that users miss targets to the bottom and right of targets, in smartphone devices [Henze11, Henze12]. Taking this into account, we deviated the real touch area of each key 10% of the key's height to the bottom, and 10% of the key's width to the right in our implementation (Figure 1d). Note that visually for the user, this variant is exactly the same as QWERTY. With this variant we expect users to commit less neighbor substitution errors.

4.5 Size Invisible variant

Similar to the *Width* variant already described in section 4.3, this variant increases the size of the most likely keys. However, this variant does it only internally; to the users it remains visually the same as a regular QWERTY keyboard. This approach has also been the aim of previous studies [Gunawardana10].

In our implementation, we increase the likely keys' width in 50% (25% to the left and 25% to the right) and 50% in height. We also imposed the condition of a maximum distance to the center of the key of 125% the diagonal radius of the key, so the final touch area of a likely key have rounded corners (Figure 1e). If two adjacent keys are highlighted and a touch occurs in an ambiguous area, the original boundaries of keys are preserved. With this variant we expect users to commit less substitution errors by hitting the desired key instead of the neighbor keys, since the most likely keys are internally bigger. We also want to ascertain if this improvement is significant in the overall error rate.

5. USER TESTS

To evaluate the performance of the variants we developed, we compare them to the performance of the traditional QWERTY keyboard, by asking 20 users to perform some text entry tasks. In the following sections, we depict the performed evaluation.

5.1 Participants

As we already stated, 20 participants fulfilled our user tests, 13 of which were males and 7 were females. All of the users' ages were between 19-30 years, except for a user that was 52 years old. Only 2 participants were left-handed. All participants had a college degree, except one that had a high school degree. Every single participant had previous experience with QWERTY keyboards and use it every day. Most participants (13) also use virtual QWERTY keyboards on a daily basis, 1 weekly, 4 rarely, and only 2 had never used them at all.

5.2 Procedure

At the beginning of each test, we explained each participant that the aim of the test was to evaluate each variant of the virtual QWERTY keyboard, and not the users themselves. The users were free to choose how they wanted to type: with one or two hands, with the tablet supported on the table, on the lap or on the free hand. Only 2 users typed with 1 hand while holding the tablet on the other, 13 typed with two hands with the tablet on the table and the remaining 5 typed with 2 hands and the tablet on the lap.

The test consisted in copying a sentence that was displayed on top of the screen, one at a time, and then move to the next sentence. Both required and transcribed sentences were always visible. The sentences were chosen randomly from a set of 88 sentences extracted from a Portuguese language corpus, such that no sentence was written twice per participant. These were the same sentences we used to perform the text prediction evaluation, which were extracted from another study [Nicolau13]. As we already stated, each sentence had five words with an average size of 4.48 characters and a minimum correlation with the language of 0.97. In order to avoid different correction strategies by the users, the delete key was removed, so users were not allowed correct errors. Participants were instructed to continue typing if an error occurred.

Before the evaluation, users were allowed to try each keyboard variant for two minutes so they could familiarize themselves with the several variants. In this training phase, users were only allowed to try the variants that had visual changes. Therefore, users were not aware of the *Shifted* and the *Size Invisible* variants.

On the evaluation phase, participants were instructed to type the sentences as quickly and as accurately as possible. Each user was asked to insert 5 sentences for each variant, where the first was still a trial and would not count to the results. The order in which each variant was evaluated was random, so that the possible effect of a user getting better at typing along the test would not benefit the results of the later tested variants. Before the test began, the users were informed that they would perform tests on 2 more variants that were only slightly different from QWERTY. And in the evaluation itself, users did not know whether they were using the *Shifted* or the *Size Invisible* variants, or even the traditional QWERTY. This way, we ensured that their typing pattern was not influenced. In the end, users were asked to answer a survey with some demographic data, as well as satisfaction regarding each variant. The whole process took about 30 minutes per user.

5.3 Apparatus

A Samsung ATIV Smart Pc Pro was used in the user study. Each key had 2 cm of width and 1.5 cm of height. Visually, there is a space of 0.2 cm between keys, horizontally and vertically. However, our implementation does not allow pressing between keys - each touch is always assigned to a key. All participants' actions were logged through our evaluation application, so posterior analysis could be performed.

6. RESULTS

In this chapter we try to understand how users responded and performed to the several variants we developed. In the first section we will scrutinize the results logged by the evaluation application, focusing on typing speed and the types of errors. Then, in section 6.2, we will take into account the answers to the satisfaction questionnaire.

6.1 Typing Performance

While the users were performing the tests, data regarding the touch positions and time was automatically recorded. This allowed us to calculate the typing speed for each variant, as shown in the boxplot in Figure 2.

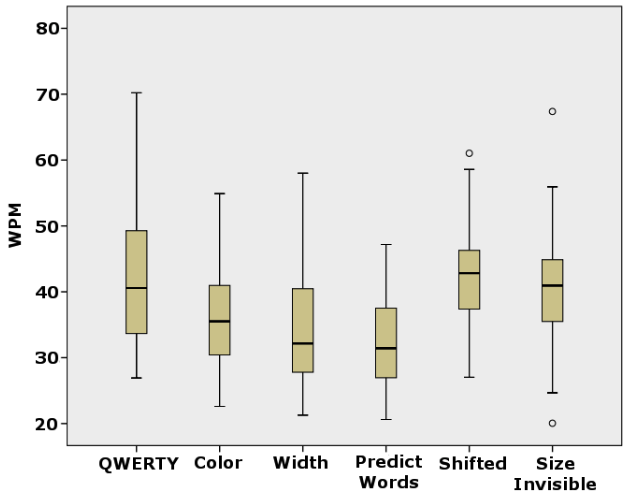

Figure 2: Typing speed of each variant.

A repeated measures ANOVA revealed a significant main effect on each variant on typing speed ($F(5, 90) = 18.787$, $p < 0.001$). Bonferroni post-hoc tests showed significant differences between QWERTY and *Color*, *Width* and *Predict Words* variants, meaning that users type significantly slower in these 3 modalities in comparison to QWERTY.

This result was somewhat expected for the *Color* and *Width* variants, since they try to avoid errors by drawing attention to different visual elements. This, in turn, may slow down the whole process of inputting text through the virtual keyboard. However, we think that these two variants may increase the typing speed of users that are not completely familiarized with the QWERTY layout, since it could help them locate letters faster. Further work is required to prove this assumption.

Regarding the *Predict Words* variant, despite the fact that users can save some keystrokes by accepting a full word in the suggested words' list, this variant was significantly slower when compared to typing on the traditional QWERTY keyboard. It seems that the cognitive effort and time required to constantly check the suggestions list does not make up for the saved keystrokes. Also, we must take into account that after typing half of the word, only 65% of the times, the word the user wants to write is in the list. This is partially because Portuguese is a highly inflected language, and thus it is very difficult to correctly guess the verbs' conjugation. We can increase the success rate by increasing the complexity of the prediction algorithm or by suggesting more than 4 words, but the later would have the drawback of taking even more time for users to read the suggestions' list. However, even with these improvements, we don't believe that the *Predict Words* variant can overcome the traditional QWERTY in typing speed, at least in the conditions we performed the tests. Most of our users were writing with both hands and multiple fingers, and in this case it is very fast to type on a QWERTY keyboard, even on a virtual one. In a situation in which users can, for example, write only with their forefinger, a feature like this could increase the speed. However, these are only speculations and a more detailed study should be performed in order to confirm this hypothesis.

As expected, there were no significant differences between the typing speed of the traditional QWERTY and the *Shifted* and the *Size Invisible* variants, since there were no visual differences causing entropy or another attention demanding feature.

To calculate the errors introduced by users in each variant, we used the Levenshtein distance between the typed and the expected sentence. The boxplot in Figure 3 shows the percentage of errors by variant. As we can see, all modalities slightly improved the overall quality of the typed sentences, since the error average is highest on QWERTY. To confirm if these improvements were statistically significant, and since our dependent variable was not normally distributed for each category of the independent variable, we used a Friedman test. Results showed that the p-value was $X > 0.05$, which means there are no statistically significant differences.

However, these results regard all types of errors, i.e., insertions, omissions, neighbor substitutions and cognitive substitutions. The latter two differ in that the neighbor substitutions errors occur when the user intended and aimed for the expected key, but missed it and ended up pressing on a neighbor key. The cognitive substitutions are errors where users simply press a key that is neither the expected one nor a neighbor key, due to a cognitive fault. The *Shifted* and *Size Invisible* variants only aim to correct the neighbor substitution errors.

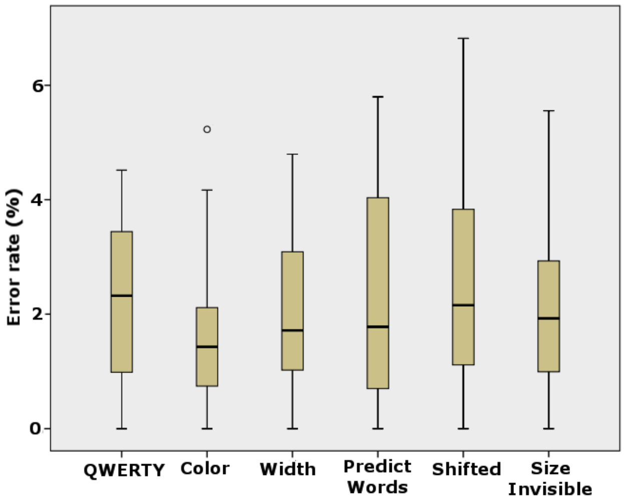

Figure 3: Error percentage of each variant.

To further analyze the error rate results, we classified each error committed by the users. As we can see in the chart in Figure 4, the neighbor substitution errors are indeed the most common errors users commit when typing on virtual keyboards. To have more data and thus more precise results, we considered the data from QWERTY, *Shifted* and *Size Invisible* variants, not taking into account the corrections performed by the latter two (i.e., all the data was treated like typing on a traditional QWERTY).

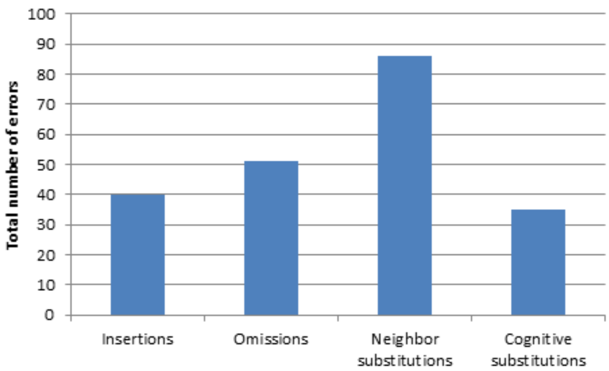

Figure 4: Frequencies of types of errors.

The *Size Invisible* variant successfully corrected 37.04% of the substitution errors, when compared to the same inputs as if the users were typing on the traditional QWERTY. However, the size of the most probable keys was set empirically. When analyzing the log data, we found that the probable keys' underlying size was increasing too much, and users were making errors because they could not correctly press the key they intended to. We then calculated the optimum size increase, which allows minimizing errors and maximizing corrections. We concluded that the optimum increase in width is 21% and 37% in height, maintaining the rounded corners. This improvement successfully corrected 62.96% of the substitution errors. Still, there are other optimizations that can even improve the success rate of this approach.

Regarding the substitution neighbor errors that still persisted in the optimized version, 11.11% of them occurred because it was in the first letter of a word and in this case the *prediction algorithm* is not working. A more advanced prediction algorithm can consider the previous typed words and, based on that information, predict the next most likely word and letter. In the other 14.81% remaining errors the algorithm had no chance of making a correction because the user had already introduced an error in that word, which means the prediction mechanism could no longer work. Only in 7.41% of errors the algorithm induced the user in error, and in the last 3.7% the algorithm did not make a correction because the intended letter was not in the most likely list.

Regarding the *Shifted* variant, it corrected several neighbor substitution errors, but it also introduced new errors due to the shift; i.e. touch events that would be correct on QWERTY, were not on the *Shifted* variant. Overall, an improvement of only 13.51% of the neighbor substitution errors was found, when compared to the same inputs as if the users were typing on a QWERTY. This mediocre result happened because a bottom-right pattern was not found neither for all the keys nor all the users. It was also because we were shifting the keys too much. After analyzing the log data, we concluded that the optimum horizontal shift is 7% of the keys' width and the vertical shift is 6% of the keys' height. This improvement allows correcting 48.65% of the neighbor substitution errors. This is partially because the most consistently frequent substitution errors occur on the bottom-right side of the 'a' key (35% of all neighbor substitution errors in the *Shifted* variant tests). And since the 'a' is the most frequent letter in the Portuguese alphabet, it represented a great deal of corrections.

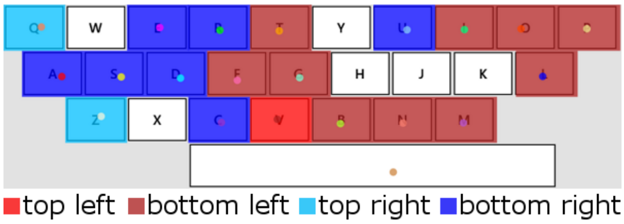

Figure 5: Average deviations of all users.

When looking at the average center of touches (Figure 5), we can see that there is an overall tendency to touch on the bottom-right side of the keys in the left side of the keyboard, and on the bottom-left side of the keys in the right half of the keyboard. Our result contradicts the results from other studies [Henze11, Henze12], because it shows that tablets will not benefit from the shift usually used in small touch devices (e.g.: smartphones). Indeed, when looking at the deviation from the center of the key of each user, we found that this deviation is strongly user-dependent. Furthermore, the same user can present different touch typing patterns, depending on the hand posture used for typing [Yin13]. Therefore, an adaptive model that recognizes various hand postures and constantly updates the center of each individual key seems to be the best solution to correct the neighbor substitution errors, without resorting to a predictive system.

Despite that none of the variants we developed showed significant improvements regarding the quality of the typed sentences, we performed a t-test between the

QWERTY and the *Color* variant, since the *Color* variant was the one with least errors. The t-test confirmed there is a statistically significant difference between these variants ($t(17) = 3.151$, $p = 0.006$). This means that, despite all the users were already familiarized with the QWERTY layout, they were committing less errors with this variant. The *Color* variant improved mostly on insertion and omission errors. However, regarding insertions, similar results were obtained in other variants, and therefore this improvement may not be significant. Indeed, the insertion is an error that is cognitively related, since users are typing fast and this increases the probability of inserting an undesired character.

An omission can also be originated from a cognitive fault, but it is most likely to occur when users miss a key or when their finger slips (they press one key and release on another, generating no output). We noted that omissions are most frequent with the space key (47% of all the omissions are spaces, in QWERTY). It happens because this key is on the bottom of the touch screen, and sometimes the users completely miss the touch area captured by the tablet, hitting its bevel instead. For instance, those users that often missed the space bar in all keyboard variants, were able to detect they were missing it on the *Color* variant, because the space bar remained highlighted, indicating that the key was not correctly pressed. As a matter of fact, in the *Color* variant, the space omissions were lowered to only 33% of all omissions.

6.2 User satisfaction

In order to perform a subjective evaluation of the developed variants, we asked participants to answer a satisfaction survey after the experimental evaluation. The questions were only regarding the QWERTY, *Color*, *Width* and *Predict Words* variants, since users were not aware about the existence of the *Shifted* and *Size Invisible* variants.

In general, users were satisfied and found it easy to use the QWERTY, *Color* and *Predict Words* variants. Regarding the *Width* variant, users said it was difficult to use and were not happy using it. They commented that the fact that the keys were constantly changing width was visually confusing, and due to this they found harder to locate, aim and press a particular key. Some users reported that it was better not to look at the keyboard while typing, but this way they could not aim properly.

When comparing each variant to QWERTY, users said, on average, that the *Color* and *Predict Words* variants were useful. The *Width* variant obtained very disperse results in this question. However, the average answered that is was somewhat unhelpful.

Regarding the cognitive effort required to use the several variants, QWERTY was rated as the less demanding. The *Color* and *Predict Words* variants were also considered to require low cognitive effort, being the former a little less demanding. The *Width* variant was the one that required more cognitive effort. When asked about the easiness of finding a particular letter, users found it easy in QWERTY and *Color* variants, and both variants averaged the same. The *Width* variant had the worst results again; users said it was relatively difficult to find a particular letter.

Despite the fact that experimental results showed that, with the *Predict Words* variant, users are slower and make the same amount of errors as in the traditional QWERTY, they classified it as useful and easy to use. Users value the feature of being able to select a whole word from the suggestions a list, even though it worsens their typing performance. User satisfaction can be more important than efficiency, since it can dictate whether users adopt a new technology or not. It is particularly important for novice users, since they can abandon a technology simply because they dislike or miss a particular feature, even if it does not bring any advantage.

However, in general, the QWERTY averaged better than other variants in satisfaction and easiness to use, which indicates that the users prefer a visually static keyboard, as similar as possible to the physical ones.

7. CONCLUSION

In this paper we described the development and evaluation of a virtual QWERTY keyboard and 5 variants, for tablet devices. Our aim in this study was to improve the typing speed and reduce the error rate on such devices.

However, we were not able to improve typing speed; users were able to type faster with the traditional QWERTY keyboard. It was somewhat expected that, for users that already know the QWERTY layout, the *Color* and *Width* variants slowed down the typing speed, since these variants introduce visual changes that can be distracting. We thought that *Predict Words* variant had the potential to improve typing speed, since users would be able to select the desired whole word from the list of suggested words, instead of typing it. But results reveal that *Predict Words* is in fact slower than the traditional QWERTY. We assume this is because users have to divide their attention between typing and checking if the desired word is on the suggested words' list. As it was expected, no significant differences were found between traditional QWERTY and *Shifted* and *Size Invisible* variants, since all these variants remain visually static.

Regarding error rates, neither *Shifted* nor *Size Invisible* variants were able to reduce errors significantly. Still, we cannot forget that both variants are solely focused on reducing neighbor substitution errors. Taking this into account, both variants actually performed well, by correcting 48.65% and 62.96% of errors in their optimized versions, respectively. The *Color* variant has the lowest error rate of all variants, at the cost of also reducing typing speed. Still, users were generally satisfied with this variant, although they were more satisfied with QWERTY. We also expected the *Predict Words*' error rate would be lower, since users could accept a whole word without orthographic errors, which would decrease the risk of making an error. Still, this variant had similar results as the traditional QWERTY keyboard. This occurs because once the user types an error, it is impossible for the *prediction system* to suggest the desired word. The *Width* variant error rate was also fairly similar to the traditional QWERTY keyboard. However, users said it was difficult to use and were not happy using it. This result contradicts the results from study [AlFaraj09], where they achieved better results with a solution similar to the *Width* variant than with the traditional QWERTY. However, they focused on smartphones and we are focusing on tablet devices, which can justify the discrepancy.

This study answered some questions, while it raised new ones. It seems that users prefer visually static keyboards, which means that the *Width* variant does not bring many advantages. There are still some improvements that can be made to other variants. A study should be performed to understand what is the ideal number of words to be included in the suggestions' list, and thus improving the *Predict Words* variant. Regarding the *Shifted* variant, a strongly user dependent touch pattern was found. Therefore, a plausible solution is to continuously adapt the centroids of the keys, for each particular user. Since the touch pattern of a particular user may also change with different hand postures, the optimum solution should also detect and adjust the shifting based on the current hand posture. For the *Size Invisible* variant, a more advanced prediction algorithm that considers not only the letters of the current word, but also the previously typed words, can achieve better results. This way, it is possible to increase the most likely keys' size, even after a space. Also, it would be interesting to perform tests with a virtual keyboard that encompasses multiple of the developed variants.

8. ACKNOWLEDGEMENTS

This work was supported by FCT (INESC-ID multiannual funding) under project PEst-OE/EEI/LA0021/2013 and the project PAELife, reference AAL/0014/2009.

9. REFERENCES

[AlFaraj09] Al Faraj, K., Mojahid, M., and Vigouroux, N. (2009). Bigkey: A virtual keyboard for mobile devices. In *Proceedings of the 13th International Conference on Human-Computer Interaction. Part III: Ubiquitous and Intelligent Interaction*, pages 3–10, Berlin, Heidelberg. Springer-Verlag.

[Cheng13] Cheng, L.-P., Liang, H.-S., Wu, C.-Y., and Chen, M. Y. (2013). igrasp: grasp-based adaptive keyboard for mobile devices. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 2791–2792, New York, NY, USA. ACM.

[Findlater12] Findlater, L. and Wobbrock, J. (2012). Personalized input: improving ten-finger touchscreen typing through automatic adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 815–824, New York, NY, USA. ACM.

[Garay-Vitoria06] Garay-Vitoria, N. and Abascal, J. (2006). Text prediction systems: a survey. *Univers. Access Inf. Soc.*, 4(3):188–203.

[Goodman02] Goodman, J., Venolia, G., Steury, K., and Parker, C. (2002). Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces*, IUI '02, pages 194–195, New York, NY, USA. ACM.

[Gunawardana10] Gunawardana, A., Paek, T., and Meek, C. (2010). Usability guided key-target resizing for soft keyboards. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, pages 111–118, New York, NY, USA. ACM.

[Henze11] Henze, N., Rukzio, E., and Boll, S. (2011). 100,000,000 taps: analysis and improvement of touch performance in the large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 133–142, New York, NY, USA. ACM.

[Henze12] Henze, N., Rukzio, E., and Boll, S. (2012). Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2659–2668, New York, NY, USA. ACM.

[Himberg03] Himberg, J., Häkkilä, J., Kangas, P., and Mäntyjärvi, J. (2003). On-line personalization of a touch screen based keyboard. In *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pages 77–84, New York, NY, USA. ACM.

[Holz10] Holz, C. and Baudisch, P. (2010). The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 581–590, New York, NY, USA. ACM.

[MacKenzie99] MacKenzie, I. S. and Zhang, S. X. (1999). The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '99, pages 25–31, New York, NY, USA. ACM.

[Magnien04] Magnien, L., Bouraoui, J., and Vigouroux, N. (2004). Mobile text input with soft keyboards: Optimization by means of visual clues. In Brewster, S. and Dunlop, M., editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, volume 3160 of *Lecture Notes in Computer Science*, pages 337–341. Springer Berlin Heidelberg.

[Nicolau13] Nicolau, H. (2013). *Disabled 'R' All: Bridging the Gap between Health- and Situationally-Induced Impairments and Disabilities*. PhD thesis, Instituto Superior Técnico.

[Yin13] Yin, Y., Ouyang, T. Y., Partridge, K., and Zhai, S. (2013). Making touchscreen keyboards adaptive to keys, hand postures, and individuals: a hierarchical spatial backoff model approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2775–2784, New York, NY, USA. ACM.

[Zhai00] Zhai, S., Hunter, M., and Smith, B. A. (2000). The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, UIST '00, pages 119–128, New York, NY, USA. ACM.