Letter and Word Prediction for Virtual Braille Keyboard

Krzysztof Dobosz

Department of Algorithmics and Software
Silesian University of Technology
Gliwice, Poland
Email: krzysztof.dobosz@polsl.pl

Łukasz Prajzler

Department of Algorithmics and Software
Silesian University of Technology
Gliwice, Poland
Email: lukapra442@student.polsl.pl

***Abstract*—The aim of this work was to study whether word prediction can be applied to virtual Braille keyboards and can improve typing text by visually impaired smartphone user. First the keyboards' advantages and disadvantages were compared to choose one to extend with word prediction mechanism. Next the method was proposed and implemented in a form of a mobile application. Due to Braille code's structure, it was possible to apply not only word, but also a letter and dot prediction. Finally, both number of dots and letters necessary for predicting letters and words respectively within the shortest time were studied. This work verified that prediction in on-screen Braille keyboards is possible and brings noticeable benefits in typing speed. The most important observation is that just after the first dot (or blank place) it is worth searching the suggested letters and words.**

Keywords–Text entry; Braille code; Letter prediction; Word prediction; Virtual keyboard.

I. INTRODUCTION

The rapid development of mobile technology in the 21st century resulted in smartphones and tablets. Interaction began to use touch and gestures on the surface of the touch screen. This has opened up many new opportunities for users of mobile devices. Anyone can customize the touch application interface to suit their needs. Thanks to this improvement, smartphones have become an essential part of the lives of many people around the world, also for people with visual impairments. The need for an external keyboard to interact with the device has been eliminated. In the era of ubiquitous smartphones, the study of text input methods on touch screens for people with visual disabilities has become a new area of research. Many methods of entering text using Braille and implemented as virtual keyboards have been developed. Their main disadvantage is their low writing efficiency.

The aim of this thesis was to study whether word prediction can be applied to virtual Braille keyboards and how this affects the efficiency of text input methods.

The rest of this paper is organized as follows: Section II presents related work on virtual Braille keyboards, Section III analyzes the problem of prediction and describes the proposed method. Section IV assesses the effects of the applied prediction. This contribution is concluded in Section V, which also outlines the starting points for future research.

II. BACKGROUND

A very well-known layout by people with visual disabilities is that of the six cells in the Braille system. In the *BrailleType* [1], the interface consists of 6 buttons ordered in 2 columns. They are placed on the edges and corners of the screen to allow for their easier localization. Dots are chosen one by one in any order with audio confirmation. Concurrently, similar project - the *LeBraille* [2] was developed. It enriches Braille typing by audio and vibration feedback. However, interaction with virtual keyboards imitating hardware keyboards is more complex for people with visual disabilities and often requires a lot of cognitive effort from them, such as remembering the positions of the virtual keyboard keys [3], [4]. Another text entry method - the *SingleTapBraille* [5], relies on six dot interface. First tap represents a first dot in a left column. It can be performed anywhere on the screen. The coordinates of the following taps are gathered and relying on their values a Braille symbol is created. Factors which influence the relations between dots are: the coordinates of each dot, the distance between dots, the number of dots in every symbol.

Two fingers are used in *TypeInBraille* [6], where two dots type dots at the same time row-by-row. This method additionally uses swipes to confirmed letters, enter space, and delete the last entered letter. In the *SBraille* [7], the user can enter text row by row. It is enough to gesture with the thumb to make all the necessary gestures. First, the user has to divide the screen into two parts by moving the diagonal line across the screen with his thumb. Then, taps are used to select a left or right point, or gestures to indicate a full or empty line.

In the approach called *Perkinput*, the user touches the screen with three fingers at the same time, the device registers and remembers their position [8]. Next, the user enters three dots at once, column by column. First, the left column were entered, then, the right one. Another column-by-column approach, the *BrailleEasy* method for one-handed Brailing is a custom keyboard called BrailleEasy to input Arabic or English Braille codes [9]. Its implementation extends the character set to support all special characters, capital letters and numbers. Another method using three fingers is *OneHandBraille* [10]. Here, neighboring dots are replaced with swipes.

Single continuous swipe for one Braille symbol can be done instead of all taps on the screen of a mobile device [11]. Evaluation using a theoretical CLC (Curve-Line-Corner) model [12] resulted with high performance. However, this method has not been verified in practice. Similar approach represents the *EdgeBraille* [13]. It is a keyboard which gathers data from six points. These points are located on the corners and long edges of the screen. The user has to enter a continuous line connecting the points to enter a letter. Selecting the dot second time deactivates it. In the *BrailleKey* the screen was divided into four big buttons [14]. Top two are used for entering text: single tap – first row, double tap – second row, long press – third row. Left and right buttons are used for entering left and right Braille symbol columns respectively. Bottom buttons – enter and delete - are used for text editing. Space is entered by double tap on enter. However, the most efficient, practically confirmed solution is *BrailleTouch* [15]. The application interface includes six virtual buttons corresponding to Braille dots. The smartphone has to be hold horizontally and the touchscreen has to be facing away from the user. This approach provides fast, eyes-free text input, where the user's fingers hit the right places on the touch screen very accurately. Some of the mentioned methods have been adapted for use in the air using image recognition technique [16], [17]. The independence from the plane of typing is a great advantage, but a big inconvenience is fast tiredness of hands, when the user types a long text.

The second research area closely related to text input is a word prediction. Word prediction often does not have a visible effect on typing proficiency when used with a standard keyboard. However, some results of studies presented that word completion or word prediction programs would increase typing speed when used with an on-screen keyboard that also requires looking away from the source document [18]. Word prediction systems can reduce the number of keystrokes required to form a message in a letter-based AAC (Augmentative and Alternative Communication) system [19]. The work [20] suggests that predictive performance can be improved by using higher-order n-gram prediction techniques. Next one demonstrates that phrases can be offered instead of words, although the user should interprets them rather as suggestions than predictions [21]. Finally, other solution adds to static text prediction of letters and words also phonetic and similarity algorithms to reduce the user's typing error rate [22]. However, some studies indicate that the effectiveness of using word prediction software to increase typing speed may vary due to the severity of physical disability or pre-intervention typing rate [23].

The purpose of this work was to verify whether prediction in on-screen Braille keyboards is possible, and whether it brings noticeable benefits in typing speed.

III. PREDICTION FOR BRAILLE KEYBOARDS

A. Analysis

A feature which can increase typing speed and has not been studied before is letter prediction. To be able to apply a such feature, all dots should not be entered all at once. The most desired text entry method is one which enables to do that one by one. After comparison of multiple advantages and disadvantages of existing Braille text entry methods, i.e. number of gestures, fingers and hands (Table I) - the *BrailleEnter* [24] solution was chosen. In this approach, the users must tap or press on a touchscreen six times sequentially to represent a letter. The user can tap or press anywhere on the screen without any concern about the location of interactions. Tap means inactivated Braille dot, and press - the activated dots. First of all, it can be used with only one hand, what is very convenient for users, who can type even while walking and holding a cane or a dog lead in second hand at the same time. Secondly, the user does not have to localize certain points on the screen and the dots are entered in a standard order.

TABLE I. COMPARISON OF GESTURES IN EACH METHOD.

Method	*Min. gestures*	*Max. gestures*	*No. fingers*	*No. hands*
BrailleTouch	1	6	6	2
BrailleKey	2	4	4	2
BrailleEnter	6	6	1	1
BrailleType	2	6	1	1
Perkinput	2	2	3	1
EdgeBraille	1	1	1	1
TypeInBraille	2	4	3	1
SingleTapBraille	1	6	1	1
OneHandBraille	1	3	1	1
SBraille	3	3	1	1

Additionally, this method is relatively fast and has a very small error rate and both of these features can be improved by using letter prediction.

The next problem to solve is how to present suggestions for letters and words to use when typing. In standard keyboard, the user can see all the suggested words while typing, so he can immediately choose one of them, if the desired one is among the suggestions. In virtual Braille keyboards it is obviously impossible. The only one possibility is emitting the words by voice using Text-To-Speech technology.

B. The Proposed Method

In this method active dot is represented by long press, inactive by single tap as in the *BrailleEnter*. The user can tap anywhere on the screen and uses only one finger to interact with the interface. A letter is entered dot by dot, column by column. The prediction is applied just after typing the first dot, but if the user wishes, instead of verifying the all the suggestions, he can continue typing remaining dots, so the search results are more restricted and more accurate. Following gestures are used to operate the prototype application of the method:

- *single tap* – adding an empty dot to the braille pattern search sequence,
- *long press* – adding a raised dot to the Braille pattern search sequence,
- *swipe right* – reading aloud next suggestion from the list to the user according to the alphabetical order,
- *swipe left* – reading aloud next suggestion from the list to the user opposite to the alphabetical order,
- *swipe down* – accepting suggested letter or word,
- *swipe up* – clearing currently being entered letter or word.

The search results create a closed loop, so after hearing the first suggestion from the list, the user can go directly at the end of the list just by swiping left. The following algorithm presents start of the process of typing letter 'M' and possible choices to be performed by the user.

1) The user performs long press, the list is filled with letters whose Braille sign begins with raised dot. The first suggested letter is letter 'A'. The user has following choices:
    - *swipe right* - the next suggested letter is 'B',
    - *swipe left* - the next suggested letter is 'Z',
    - *single tap* - continue typing.

2) The user can swipe to reach the desired letter. However knowing alphabet order the user knows that this letter is in the middle of alphabet, so it may take a lot swipe gestures. The user performs single tap, the list is filled with letters whose Braille sequence begins with "10", so the search results list is reduced. The first suggested letter is "A". Still the user has the following choices:
   - *swipe right* - the next suggested letter is 'C',
   - *swipe left* - the next suggested letter is 'Z',
   - *long press* - continue typing.

The full path of restricting the search results list after entering consecutive dots while typing letter 'M' is presented in Figure 1. Word prediction system works the same way. After typing first four letters the system suggests a word. The words are taken from the list of five thousands the most popular English words and are ordered descending, according to their occurrence frequency in English language. The user can swipe right to check next suggestion, swipe left for previous one or continue typing. The word is chosen by swiping down. In case the user is distracted or has to abandon typing for a moment, both letters and words are frequently being repeated every 10 seconds. Swiping up clears a letter which is currently being entered. If no letter is currently being entered, swipe up clears the initial word sequence.

IV. Evaluation

A. Procedure

The evaluation procedure consisted of three parts. First, the number of gestures needed to enter each letter of the English alphabet was counted. Then, using the selected letter prediction method, typing of single words was tested. Finally, when both the letter and word prediction methods were chosen, final tests using a pangram were performed.

Before proceeding with testing, theoretical considerations were performed. Relying on each gesture duration and their amount in each investigated method, average time for every method was estimated after several trials:

- tap - 0.104 s.,
- long press - 0.771 s.,
- swipe - 0.114 s.,
- double tap - 0.215 s.,
- letter speech - 0.278 s.,
- word speech - 0.543 s.

Next, total number of gestures necessary for typing each letter using each method was calculated introducing a measure - GPC (Gestures Per Character). One sentence was selected to perform the final test. This sentence was a pangram that contains every letter of alphabet to perform the most reliable evaluation. The pangram was tested using following methods:

- basic *BrailleEnter* method,
- letter prediction applied after 1^{st} dot,
- letter prediction applied after 2^{nd} dot,
- letter prediction applied after 3^{rd} dot,
- using both letter after the 1^{st} dot and word prediction.

The set of data for letter prediction was just a set of every letter of English alphabet and its Braille sign representations.

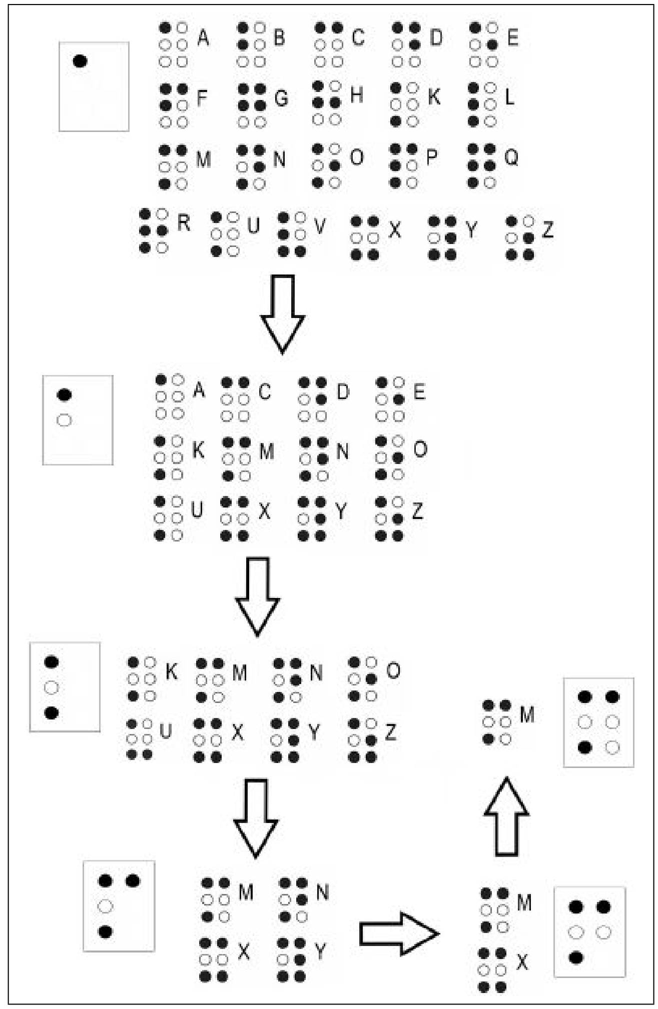

Figure 1. Steps for limiting a character set.

The list of 5000 the most popular English words was taken from the Corpus of Contemporary American English [25]. A sentence used for the final performance was a very popular English pangram *"The quick brown fox jumps over the lazy dog"*.

B. Letter Prediction

First average typing time for each letter was measured. Occurrence of each letter in the sentence was counted and multiplied by the average typing time of the letter. Swipe gestures for reviewing words suggestions and average time of letter speech, as well as double tap for entering spaces were added to the estimations. Then, all types of gestures and actions which occur in the sentence were counted. The numbers of each of them in each of the approaches were multiplied by the average timing calculated. Then, the results were summed up. Comparison of duration of each approach to the prediction is presented in Table II. The number of swipe gestures is always equal the number of suggested letters. In both approaches to the estimation, the best result was achieved, where prediction is applied after 2nd dot. Slightly worse scores obtained method, where prediction is applied after the 3rd dot. Next, estimated values were verified using the research tool in a form of mobile

TABLE II. AVERAGE TOTAL GESTURES TIME[S] AND GPC VALUE.

Gesture	$1^{st}dot$	$2^{nd}dot$	$3^{rd}dot$	$4^{th}dot$	$5^{th}dot$	*BrailleEnter*
tap	0.02	0.07	0.11	0.15	0.21	0.28
press	0.62	1.04	1.48	1.96	2.31	2.55
swipe	0.62	0.34	0.26	0.19	0.14	-
speech	1.52	0.83	0.63	0.47	0.34	-
time	2.78	**2.28**	2.48	2.77	3.00	2.83
GPC	6.46	**5.00**	5.27	5.69	6.23	6.00

application. After performing the experiments, the comparison of duration of letter prediction methods and *BrailleEnter* method clearly proves that application of letter prediction can significantly increase typing speed. The results and average times for each letter and method are presented in the Table III. The best time for each letter was bold.

TABLE III. AVERAGE TIMES OF LETTER TYPING WITH PREDICTION.

Letter	$1^{st}dot$	$2^{nd}dot$	$3^{rd}dot$	$4^{th}dot$	$5^{th}dot$	*BrailleEnter*
A	**0.61**	1.22	2.21	3.14	3.88	3.70
B	**1.04**	1.90	2.97	3.51	3.79	4.10
C	**1.47**	2.54	3.91	3.40	3.60	4.08
D	**2.05**	2.05	4.34	4.44	4.52	3.98
E	**2.97**	4.21	3.81	4.13	4.47	3.71
F	4.62	**3.47**	4.24	3.95	4.25	4.52
G	4.54	**3.84**	5.00	4.61	5.06	4.90
H	4.52	**3.91**	4.43	4.41	5.07	4.43
I	**0.49**	1.32	2.68	3.57	3.87	4.05
J	**0.93**	2.18	3.95	4.40	4.41	4.68
K	5.67	5.02	3.15	**3.13**	3.87	4.05
L	6.92	6.11	**3.81**	4.02	5.10	5.22
M	7.04	6.14	4.12	**3.82**	4.25	4.89
N	6.21	5.71	5.15	**4.51**	4.78	5.17
O	4.97	5.24	6.02	4.09	4.14	**3.98**
P	**4.76**	6.05	4.85	5.22	5.96	5.29
Q	**4.47**	4.94	6.28	5.94	7.53	5.92
R	**3.90**	4.69	6.47	5.20	5.73	5.26
S	**1.31**	3.33	3.70	4.43	5.11	4.99
T	**1.84**	3.93	4.67	5.66	6.18	5.39
U	**3.44**	5.09	7.10	4.69	4.96	4.39
V	**2.95**	3.27	5.01	5.60	6.36	5.26
W	**0.93**	2.61	3.93	4.70	6.22	5.11
X	**2.46**	3.56	5.48	6.24	5.23	5.27
Y	**1.67**	2.86	4.54	4.86	6.23	5.46
Z	**1.20**	2.81	3.89	5.16	5.74	4.58
Avr	**3.19**	3.80	4.45	4.49	5.01	4.70

The best typing speed was obtained for a method, where letter prediction is applied just after typing the first dot. These results are confirmed by analysis of normalized data, where the results are even better. However, not for every letter this method appeared to be the best. Time necessary for typing letters 'K', 'L', 'M', 'N', 'O' was higher than for the remaining letters, what is more for the first 4 of them this method appeared to be the most time consuming. The list of predicted letters was quite long due to relying only on the first dot while creating it and these letters were in the middle of the list. That is the reason why they required more time and more swiping gestures and the total number of required gestures to entering them was the highest from all the gathered data and varied from 10 to 12. Letter 'O' as the only one from the whole alphabet was entered the fastest using only the basic *BrailleEnter* method.

Table II does not reflect exactly the results of total direct letter time measurements, presented in Table III. First of all, direct letter times are much greater than the ones derived in calculations. Secondly, the order of best timing methods is not the same as in the letter measurements. However, after taking into account some circumstances, it can be seen that the results are reliable. The divergences are caused by the fact that in gestures measurements only the time when finger touches the screen was taken into account. While typing a letter, there are multiple factors which can influence typing speed, among them are: duration of putting up and down a finger, temporarily slowdown of the device, human error. After taking into consideration these factors it can be seen that the results are similar.

Finally, the task was to choose the best letter prediction method for further tests. This application does not assume setting fixed number of dots necessary to predict letter, only the minimal number. The results show that the highest typing speed is obtained when applying letter prediction already after the first dot. However, due to the fact that some letters achieved better results after higher number of dots, it may be more efficient, if in case of these letters more dots will be typed. And here can be seen that a big benefit for potential users would be freedom of choice. For users who are familiar with Braille code, this method would be definitely better choice. On the other hand, there are many visually impaired people who do not know the Braille code. Applying letter prediction after the first dot allows them for entering text relatively fast without the necessity of learning in details the Braille code. What is more, this method decreases the possibility of mistyping or entering a wrong letter a lot. If the first dot is incorrect, it can be easily and quickly corrected.

C. Word Prediction

The same as with letter prediction, this part began with theoretical analysis. In this case again, knowing number of each gesture in each letter and duration of each gesture, estimated typing time of each word was calculated. However, taking into account the differences between time calculation and experiments results, and knowing the average duration of typing each letter, second calculation which instead of analyzing each letter components in detail uses already measured average time for each of the typed letter.

The next task was to estimate typing time of one word. To achieve that average number of each action for each method was multiplied by the average duration of each action. The results are presented in Table IV.

TABLE IV. AVERAGE DURATION OF ACTIONS IN EACH METHOD OF WORD PREDICTION.

Gesture	*3 letters*	*4 letters*	*5 letters*	*6 letters*	*BrailleEnter*
tap	-	0.03	0.04	0.07	0.16
press	2.31	2.89	3.57	4.11	5.78
swipe	5.53	3.61	4.08	4.45	6.27
letters	6.81	7.96	9.35	10.33	15.29
words	13.03	1.63	1.15	1.00	-
total	27.69	**16.11**	18.18	19.95	27.50
estimation	29.51	**18.35**	20.83	22.47	31.81

The results are quite similar to the GPC results. For instance, it can be seen that average scores are the best for word prediction applied after 4th letter (one before last row). Afterwards, the average word entry duration was estimated using already measured letter entry time. Number of each letter occurrence in each method for all words was calculated and multiplied by the certain letter average duration. These results are presented in the last row of Table IV. After analysis of GPC metric and both types of time estimation it can be seen that the results are very similar to each other. Word prediction applied after 4th letter appeared to be the best.

D. Final evaluation

The final test was also preceded with theoretical considerations and estimation. First the pangram sentence is analyzed using GPC metric, than typing time is estimated using two approaches. The same as with word evaluation, first approach uses average duration of each gesture, the second uses estimated duration of the gestures and measured letter entry duration. For evaluation purposes it was assumed that only first 6 suggested words are checked. If there are no hits among them, the whole word is being typed. To perform estimation the most optimistic scenario was taken into account and it was assumed that all words are found in the database and they are correct just with the first word proposal. The mobile research tool was supplied with another feature – entering space after double tap, to allow typing whole sentences.

The first estimation consisted in counting all types of gestures and actions which occur in the sentence (Table V). The best results are obtained for the approach with letter prediction applied only after the second dot. The worst score again belongs to letter prediction after the first dot. The difference is very big here, it is over 20 seconds.

The second estimation consist in using measured average typing time for each letter presented in Table III. Occurrence of each letter in the sentence was counted and multiplied by the average typing time of the letter. Swipe gestures for reviewing words suggestions and average time of word speech, as well as double tap for entering spaces were added to the estimations. In this case the best results were obtained for approach, where both letter and word prediction are applied (one before last row in the Table V).

Measurements prove that application of both letter and word prediction can significantly improve typing speed in Braille virtual keyboards on touchscreen devices. Approach with letter prediction applied after the 1st dot and word prediction applied after the 4th letter obtained the best result(last row in the Table V). That is 20.15s (2.46 WPM). Slightly worse result was obtained for approach, where only letter prediction after the first dot was applied. Basic *BrailleEnter* method scored the worst result (284.9s equals 1.77 WPM). Application of both letter and word prediction improved typing speed by almost 80 seconds what is a very good result, especially that this score is for only one sentence.

Some remarks were made during the experiments. First, it should not be forgotten that letter prediction evaluation showed that some letters were typed faster after applying prediction after higher number of dots than one. If this piece of knowledge was taken into account, the typing speed for some letters could be increased by 1 to over 3 seconds. Secondly, in case of used pangram only 3 words were long enough to apply the word prediction. What is more, each of these words was only 5 letters long. Two of these words were found in the database and suggested as the firsts on the lists. In case of the third word, the word found in the database was exactly the same as the first 4 typed letters, and there was necessity to type additional letter "s" at the end of the word to create a plural form. It shows that the word prediction did not influence in this case the typing speed a lot – the difference in time between two the fastest approaches is only about 9 seconds.

TABLE V. ESTIMATED DURATION OF ACTIONS FOR THE PANGRAM.

Gesture	1^{st}dot	2^{nd}dot	3^{nd}dot	words	*BrailleEnter*
tap	0.06	2.50	3.95	0.52	10.09
press	22.36	35.47	51.66	20.82	87.12
double tap	1.72	1.72	1.72	1.72	1.72
swipe	23.48	13.00	9.92	21.20	-
letters	52.27	31.69	24.19	50.87	-
words	-	-	-	-	-
total	105.46	**84.37**	91.43	96.76	98.93
2nd estimation	119.24	142.21	165.72	**112.05**	162.80
measurement	214.25	246.25	255.02	**205.17**	284.90

V. Conclusion and Future Work

The aim of this work was to study whether typing speed on virtual Braille keyboards can be improved by using letter and word prediction algorithms. First, the existing Braille text entry method worth to be improved was selected. Afterwards, the prediction mechanism was chosen. Several variants for letter and word prediction were designed and implemented as complete research tool in a form of mobile virtual keyboard. Next letter and word prediction were analyzed, including gestures count, different gestures types and their duration. Obtained results confirmed that prediction in on-screen Braille keyboards is possible and brings noticeable benefits in typing speed. The most important observation is that just after the first dot (or blank place) it is worth searching the suggested letters and words. Obtained result equals 2.46 WPM is better than 1.77 WPM for reference *BrailleEnter* method.

In the future, the experimental virtual Braille keyboard can be extended with multiple additional features to increase typing speed even more. For instance, there are many advanced word prediction algorithms which could be applied to both improve accuracy of predicted words.

Acknowledgment

This work was co-financed by SUT grant for maintaining and developing research potential.

References

[1] J. Oliveira, T. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves, "Brailletype: unleashing braille over touch screen mobile phones," in IFIP Conference on Human-Computer Interaction. Springer, 2011, pp. 100–107.

[2] A. R. Façanha, W. Viana, M. C. Pequeno, M. de Borba Campos, and J. Sánchez, "Touchscreen mobile phones virtual keyboarding for people with visual disabilities," in International Conference on Human-Computer Interaction. Springer, 2014, pp. 134–145.

[3] H. Tinwala and I. S. MacKenzie, "Eyes-free text entry on a touchscreen phone," in 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH). IEEE, 2009, pp. 83–88.

[4] M. N. Bonner, J. T. Brudvik, G. D. Abowd, and W. K. Edwards, "No-look notes: accessible eyes-free multi-touch text entry," in International Conference on Pervasive Computing. Springer, 2010, pp. 409–426.

[5] M. Alnfiai and S. Sampalli, "Singletapbraille: Developing a text entry method based on braille patterns using a single tap," Procedia Computer Science, vol. 94, 2016, pp. 248–255.

[6] S. Mascetti, C. Bernareggi, and M. Belotti, “Typeinbraille: a braille-based typing application for touchscreen devices,” in The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility, 2011, pp. 295–296.

[7] S. Lee, J. S. Park, and J. G. Shon, “Sbraille: A new braille input method for mobile devices,” in Advances in Computer Science and Ubiquitous Computing. Springer, 2017, pp. 528–533.

[8] S. Azenkot, J. O. Wobbrock, S. Prasain, and R. E. Ladner, “Input finger detection for nonvisual touch screen text entry in perkinput,” in Proceedings of Graphics Interface 2012, 2012, pp. 121–129.

[9] B. Šepić, A. Ghanem, and S. Vogel, “Brailleeasy: One-handed braille keyboard for smartphones.” Studies in health technology and informatics, vol. 217, 2015, pp. 1030–1035.

[10] K. Dobosz and M. Szuścik, “Onehandbraille: an alternative virtual keyboard for blind people,” in International Conference on Man–Machine Interactions. Springer, 2017, pp. 62–71.

[11] K. Dobosz and T. Depta, “Continuous writing the braille code,” in International Conference on Computers Helping People with Special Needs. Springer, 2018, pp. 343–350.

[12] X. Cao and S. Zhai, “Modeling human performance of pen stroke gestures,” in Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, pp. 1495–1504.

[13] E. Mattheiss, G. Regal, J. Schrammel, M. Garschall, and M. Tscheligi, “Dots and letters: Accessible braille-based text input for visually impaired people on mobile touchscreen devices,” in International Conference on Computers for Handicapped Persons. Springer, 2014, pp. 650–657.

[14] N. S. Subash, S. Nambiar, and V. Kumar, “Braillekey: An alternative braille text input system: Comparative study of an innovative simplified text input system for the visually impaired,” in 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI). IEEE, 2012, pp. 1–4.

[15] M. Romero, B. Frey, C. Southern, and G. D. Abowd, “Brailletouch: designing a mobile eyes-free soft keyboard,” in Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, 2011, pp. 707–709.

[16] K. Dobosz and K. Buchczyk, “One-handed braille in the air,” in International Conference on Computers Helping People with Special Needs. Springer, 2018, pp. 322–325.

[17] K. Dobosz and M. Mazgaj, “Typing braille code in the air with the leap motion controller,” in International Conference on Man–Machine Interactions. Springer, 2017, pp. 43–51.

[18] D. Anson et al., “The effects of word completion and word prediction on typing rates using on-screen keyboards,” Assistive technology, vol. 18, no. 2, 2006, pp. 146–154.

[19] K. Trnka, J. McCaw, D. Yarrington, K. F. McCoy, and C. Pennington, “User interaction with word prediction: The effects of prediction quality,” ACM Transactions on Accessible Computing (TACCESS), vol. 1, no. 3, 2009, pp. 1–34.

[20] G. W. Lesher et al., “Effects of ngram order and training text size on word prediction,” in Proceedings of the RESNA’99 Annual Conference. Citeseer, 1999, pp. 52–54.

[21] K. C. Arnold, K. Z. Gajos, and A. T. Kalai, “On suggesting phrases vs. predicting words for mobile text composition,” in Proceedings of the 29th Annual Symposium on User Interface Software and Technology, 2016, pp. 603–608.

[22] R. d. S. Gomide et al., “A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques,” Research on Biomedical Engineering, vol. 32, no. 2, 2016, pp. 176–198.

[23] J. Tumlin and K. W. Heller, “Using word prediction software to increase typing fluency with students with physical disabilities,” Journal of Special Education Technology, vol. 19, no. 3, 2004, pp. 5–14.

[24] M. Alnfiai and S. Sampalli, “Brailleenter: A touch screen braille text entry method for the blind,” in ANT/SEIT, 2017, pp. 257–264.

[25] M. Davies. Corpus of contemporary american english, word frequency data. [Online]. Available: https://www.wordfrequency.info/