International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 6 (2015) pp. 15183-15195

http://www.ripublication.com

See Through –Design and Implementation of An Innovative Mobile Keypad For The Visually Challenged

Gowtham Raj P
+918220007008
gowthampoy1994@gmail.com
M Anirudh Srinivas
+919566226855
anirudhmadhavan@ymail.com
Arjun K
+917598660199
arjunk1993@gmail.com
Bindu K R
Assistant Professor (Sr. Gr.)
+919489921989, j_bindhu@cb.amrita.edu
Dept. of Computer Science and Engineering,
Amrita School of Engineering, Amritanagar Post
Ettimadai, Coimbatore - 641112, Tamil Nadu, India

Abstract

In today's world, where mobile phones are as essential as food and shelter, we present an innovative text-entry method for the visually challenged people. Existing keypads are not suited for the visually challenged since it involves a lot of memorization and the process is time consuming.

We propose to use the design of a basic 3X4 keypad and modify it to make it user-friendly for the visually challenged. We plan to use a new method, which divides the alphabets into groups and generates a matrix like structure. Using four keys for navigation, the user can enter any alphabet by traversing through this matrix. The other keys would be used for miscellaneous functions to further assist the users. We also plan to use an enhanced audio feedback system, automatic word completion and task aware dictionaries for enhancing the functionalities. We have implemented the prototype of this keypad as an Android application and evaluated the same among target usersto set the foundation for future research and possible commercial implementation.

Keywords: 'Visually Challenged', 'Text-Entry', 'Audio Feedback', 'Dictionary', 'Automatic Word Completion', 'Android',' Task Aware Dictionaries'

Introduction

In this digital age, mobile phones have become an integral part of our everyday lives and it is now second nature for us to turn to them at the drop of a hat. However, they strongly rely on visual feedback for effective operation. We feel that modern mobile phones should be versatile enough to cater to the needs of each one of us and our focus is on empowering the visually challenged in this regard.

In this paper, we have discussed the different keypads that exist in the market and the challenges faced by visually challenged people when they attempt to use them. We have also touched upon visually impaired focused approaches and why they are not solving the problem at large. We have developed a new text-entry interface that attempts to overcome most of the problems, thereby making the usage of regular mobile devices possible, without incurring additional costs. This will ensure that users input words at an acceptable rate, which in turn will enhance their capabilities.

We have discussed and developed the prototype of the design of our suggested text-entry interface, wherein visually challenged users are expected to be at ease. We have developed an Android application to simulate our model and we have formulated this paper with our work. This paper also has the technical details necessary to understand the Android application that we have developed to simulate our model and to extend the same, as part of future research. We have evaluated the application among target users by measuring a few key metrics. We have also provided details as to why this model might prove to be successful and how it can be extended as part of future work.

Background

Text entry is usually done using a 12 key keypad on basic mobile phones. These keys feature number keys 0-9 and two additional keys, * and #. The letters are arranged in alphabetical order, usually from keys 2-9. 0 key also performs the function of feeding space as an input.

Some of the methods which are used to obtain input from the user are elaborated below.

A. Multi-Tap System

In the Multi-Tapsystem [1], which is the most widely used system today, one or more presses are required to obtain a particular character, depending on the template of the keypad as decided by the manufacturer [Fig.1]. This precludes visually challenged people from using the mobile without suitable training and memorization of the association between the different keys and alphabets. Error recovery is also slow and difficult without feedback [4].

B. Message-Ease System

In the Message-Ease system [2], the frequency of letters used in English is studied and generalized and the alphabets are distributed accordingly, to enable easy input of frequently used alphabets. A sample distribution is shown [Fig.2]. The less frequently used alphabets are placed in the centre and the more frequently used alphabets are spread out. The more frequent keys are placed in pairs or triplets, thus making it easy to enter a frequent letter. Though this system is faster, it requires extensive memorization as the alphabets are not in order.

C. Predictive System

The predictive text entry method predicts the text that the user is going to enter using letter anticipation utilizing the frequency of alphabets or word completion by referring to the dictionary. This is usually used in conjunction with a Multi-Tap system. This system compares the sequence of keys pressed to the words in a dictionary and gives suggestions of all the possible words that match part of the typed sequence. Predictive systems can be difficult for the visually challenged to use as they won't be aware of the predictions being made. There need not be any relation between what is entered and what appears on the screen. This can prove to be disastrous when the prediction is ineffective. Without audio feedback, this system is a failure.

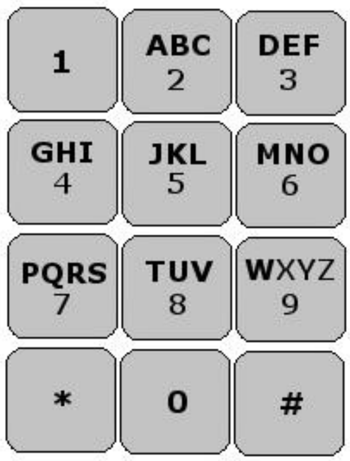

Figure 1: A sample interface design of Multi-Tap system

D. Visually Impaired Focused Approaches

Many devices have been developed to overcome the difficulties of using basic mobile phones. Devices like Braillino and Braille Tap [3] have been designed which help the visually challenged to input text and navigate their mobile devices. They are external hardware that use Braille keyboard and that can be connected to a mobile phone through Bluetooth or other connection mechanisms. It is however expensive, heavy and difficult to carry around and thus, impractical to use with a basic phone [Fig.3] [4].

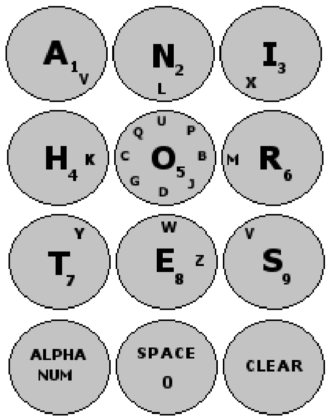

Figure 2: A sample distribution of Message-Ease System

It is even more challenging for visually challenged people to interact with touch screens as familiar input methods, such as keyboards, are replaced by their onscreen counterparts [5]. The lack of physical feedback and the fact that these devices are devoid of physical buttons proves to be the biggest hindrance for the users. Further, the interface constantly changes from screen to screen, making it hard to navigate and access the desired content. The aforementioned problems make touch screen text-entry a major challenge for a visually challenged person.

Figure 3: A model of a Braillino

Visually impaired subjects we interviewed reported that they tend to not use mobile phones. This poses a serious problem to the balance in our society. The problems associated with limited visual feedback on mobile text-entry have been evaluated and widely discussed [6] and this led to our search for an easier and inexpensive alternative.

Related Work

There has been a lot of research going on in the field of eyes-free text entry in the last decade. A lot of hardware and software solutions have been proposed. However, the hardware solutions are expensive while the software solutions impact the performance. Most of the software solutions are applicable only for touch screen devices.

Braille Touch is an eyes-free text entry technology for touch screens. It is a Braille keyboard which is available as an application on iTouch. Brailler is a single-touch text-entry system which was developed for touch screen devices [5]. Brailler allows the visually challenged user to enter text as if he was writing Braille using the traditional 6-dot matrix code.

Many new tools have attempted to change the input method in touchscreen phones from the traditional QWERTY keypad to Braille input keypads and have focussed on the design of the same [7]. Aside from Braille input, Fleksy [8] is an innovative application designed for iPhone users to input text through extensive prediction. The taps made by the users are recorded and prediction is carried out on the same. Letter Scroll [9] is an innovative technique which attempts to enable users to input text by using a mouse wheel to manoeuvre over a sequence of characters and a button for character selection.

Even though these tools are compelling and the software solutions are cheaper than the hardware alternatives, they have certain performance issues which limit the accessibility for visually challenged users. A common weakness for many of these solutions is the difficulty in changing already entered text. Moreover, the focus has shifted to developing applications which will change the way users interact with the mobile devices. A fundamental shift in the way users input text is the need of the hour.

Design

We worked on the design of a navigation method which will involve less memorization and more of common knowledge which might be considered as a prerequisite to use this system. We decided to play around with English alphabets and split them into groups with each vowel considered as the starting point of the group. We then decided to form a matrix with the first column having all the vowels and the subsequent columns consisting of the alphabets succeeding the vowel in its particular row. The exact structure is described in the figure [Fig.4]. The essence of this system lies in the ability to split English alphabets into groups according to the position of the vowels. This is an extension of the NavTap system [10] which has been proposed.

Next, we had to find a way to traverse this matrix. We realized that the biggest challenge for the users was the tediousness involved in using 12 keys for inputting text. We decided to do away with that system and use only 4 keys for navigation. In basic phones, there is usually a mark around the 5 key and thus, the keys horizontally and vertically adjacent to 5 can be used for easy navigation. The navigation can thus be done using 2, 4, 6 and 8 keys and it can also be cyclic for faster typing [10]. For example, pressing down (2) and left (4) will take you to H. The user then presses the select key (5) to confirm the input of the alphabet H. We assign a specific key to input space and punctuations. Even though this does not guarantee faster performance, it does not necessitate a need for the users to memorize the position of alphabets since there is no longer any association between the keys and

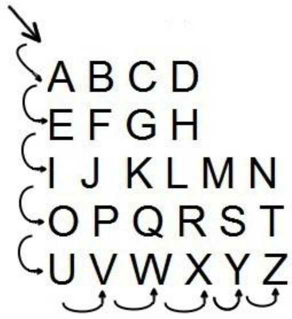

Figure 4: An example of navigation from A to Z

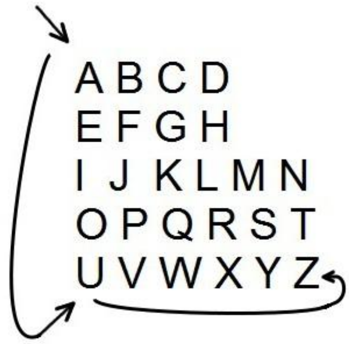

Figure 5: An example of cyclic navigation from A to Z

alphabets. Any person with common English knowledge can use this system. If used effectively, this system can reduce the time taken to input a particular alphabet.

We also propose to use three different systems in conjunction with this system.

A. Audio Feedback

Every time the user enters an alphabet, the alphabet that has been entered and the current state of the already inputted text can be given as audio feedback. This can help in error recovery as the user can enter an erase button (should be configured) to go back.This system also solves one major problem that other alternatives face in changing previously entered text. Since a single key press (9) is used for removing text, alphabet by alphabet, users can press the key and obtain audio feedback at every stage. This makes it easy for them to alter inputted text.

B. Automatic Word Prediction

To increase texting speed, auto word suggestion is given if the user presses '*' on the keypad and punctuations can be cycled through using the number '1'. As soon as * is pressed, the word is predicted based on user statistics and the English dictionary and audio feedback of the predicted word is given out. If the user presses the select button (5), the existing word is replaced by the predicted word. If the user presses the next button (*), the next word in the sequence of predicted words is displayed and read out.

User can also press 3 to repeat the last suggested word. '#' key can be used to toggle between numbers and alphabets whereas key 7 can be used to read the contextual meaning of the selected word thus, encouraging the user to use complicated words when necessary. This calls for a highly efficient dictionary system and prediction system.

The rest of the keys in the keypad can be used for miscellaneous purposes according to the needs of the user. This gives us scope to add additional functionality to the keypad. The functionalities that we have proposed are tabulated below.

Table 1: Keys and Their Functions

Keys	**Functions**
*	Next prediction word
#	Change between uppercase, normal and numeric
1	Cycle through punctuations
2	Traverse vertically
3	Repeat the last suggested word
4	Traverse horizontally
5	Commit the suggested word
6	Traverse horizontally
7	Read the contextual meaning of the selected word
8	Traverse vertically
9	Erase
0	Space

C. Task Aware Dictionaries

Task aware dictionaries help us to predict the word depending on the context. For example, if the user is typing a name in the contacts field, the prediction system will not suggest any word. At the same time, if the user starts entering a name in the message recipients field, the system proposes the names available in the contacts. Depending on the task that the user is performing, the prediction system suggests words accordingly. This system can be integrated with our prediction system to offer additional functionality to our mobile interface. Since visually challenged users need as much assistance as possible, this system will ensure that the word prediction is sensible [4].

Our design allows the traversal of the matrix to be cyclic [Fig.5]. This drastically brings down the number of keystrokes required to input text. For example, to input 'U', the user needs to press the down arrow once and to input 'Z', press left from 'U'. Once users get accustomed, this feature can be put to use to great effect for easy text-entry.

Based on the number of users and the availability of the devices, our approach can be easily implemented and used without any requirement for external hardware as it is a simple software implementation. Even though the users might face difficulty in adapting to the system initially, proper orientation will lead to increased comfort while using mobile phones.

Our system attempts to add extra features and integrate disparate systems to support the design proposed in NavTap for increased comfort and accessibility. The essence of this system lies in its ability to be extensible by offering custom functionality through the different keys available in the keypad.

This solution is also easily extensible to other languages. Linguistic experts can determine the procedure to split the native alphabets into groups and form the character matrix. Once the matrix has been formed, the same navigation techniques can be applied to input native text.

This system can also be implemented as an application in smart phones which serves as a tool to input text which can then be copied and pasted in other applications as required.

Implementation

We decided to develop an Android application as part of this project to simulate the design of the text-entry interface. Even though this design is applicable only for basic phones, we couldn't develop a working prototype using a basic phone because of resource constraints. Since developing an Android application is inexpensive, we decided to develop such an application which will have only 12 buttons to simulate the keypad of a basic phone and a text area to display the text.

We used 12 buttons and arranged them in the form of a basic keypad. We used a text area as the text display tool. Every time the user presses a button, we invoke the On Click Listener of that particular button and perform the actions in the respective Event Handlers. We use a pointer to keep track of the current input position. In the event handler of the navigation keys, we trace the last alphabet in the text area using the pointer and change the ASCII value according to the type of navigation required. The event handler of the select key confirms the alphabet in the current pointer position and increases the value of the pointer. The event handler of the back key deletes the alphabet in the current position of the key and reduces the value of the pointer by one. When the punctuation or space button is pressed, the corresponding character is inserted in the current pointer position.

Audio feedback was achieved using Google's Text to Speech API [11]. We used the API to read out the current state of the last alphabet and the entire text in the text area. We also implemented the word prediction algorithm by keeping track of the words that the user has entered in a SQLite database which is available in an Android application by default. We keep track of the word that the user enters and the frequency of its usage in the database. We use the sequence of alphabets from the last entered space till the current pointer position for the word prediction. When a user presses the predict button (*), the words

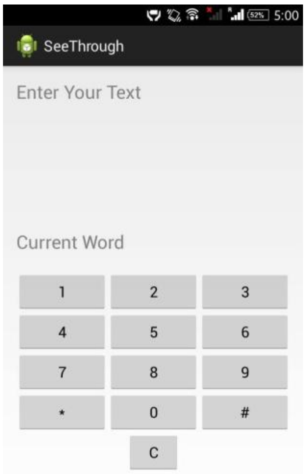

Figure 6: A screenshot of the basic design of our application

which match the entered sequence is displayed. We use machine learning algorithms like n-gram for word prediction. The order in which the words are displayed to the user depends on the usage statistics available in the database. When the user presses the confirm word key (which is different from select key), the word replaces the existing sequence, space is entered by default and the pointer moves to the next position. Once a word has been confirmed, we make suitable additions to the database and proceed. Screenshots of our application are available in [Fig.6] and [Fig.7].

This keypad simulates the design we have elaborated and this can be further prorated by implementing Speech to Text features using Google's Voice Recognition API, Task Aware Dictionaries and Auto Completion. However, this simple application was sufficient to identify the significant advantages of our system and the relative comfort that it provides to visually challenged people while typing. This application can be used for evaluation purposes.

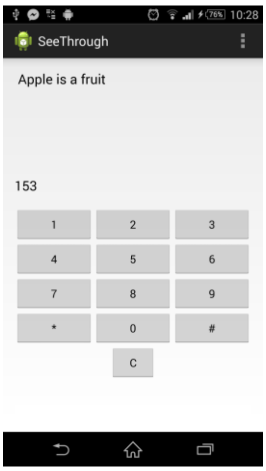

Figure 7: A screenshot of a sample text-entry using our application

Evaluation

We decided to evaluate our application with the target users for feedback. We handed over smart phones which had a fully enabled See Through application and which had a mark over the region where '5' would appear on the screen. We selected five users who had experience in feeding input through the Multi Tap system. We had three different sessions with the users and we recorded the parameters necessary to evaluate our model.

We began with recording the average keystrokes the user made for keying in random characters in Multi Tap as well as See Through. The average keystrokes made over three sessions was found to be lesser in the Multi Tap system than See Through, unsurprisingly [Fig.8]. However, it is important to highlight the fact that the average number of keystrokes reduced drastically over the three sessions with See Through [Fig.9]. As the users were getting more comfortable with See Through, they were able to form a mental map of the character matrix and type faster. This shows us that See Through will prove to be successful only after comprehensive training.

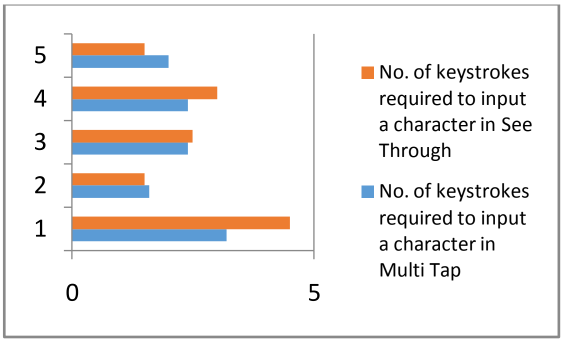

Figure 8: Comparison of the average keystrokes required

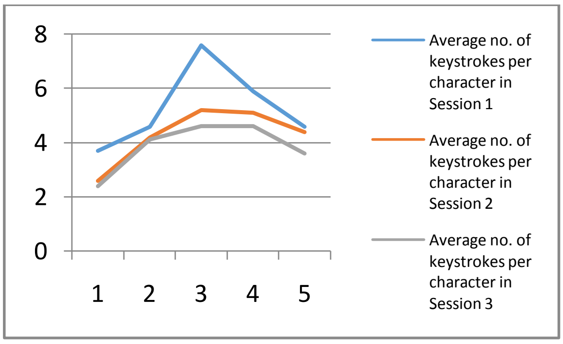

Figure 9: Comparison of the average keystrokes made over three sessions

We also recorded the number of words that a user can feed in correctly in a given time. Users were given five minutes and were asked to key in from a predefined list of words. We recorded the same both with and without the prediction system of See Through. The number of words which were incorrectly typed was negligible when See Through was used with a prediction system [Fig.10]. As was with the number of keystrokes required, the number of erroneous words typed within five minutes also reduced drastically over the three sessions [Fig.11]. This shows us that prediction and audio feedback systems play a crucial role in correct text-entry and error recovery.

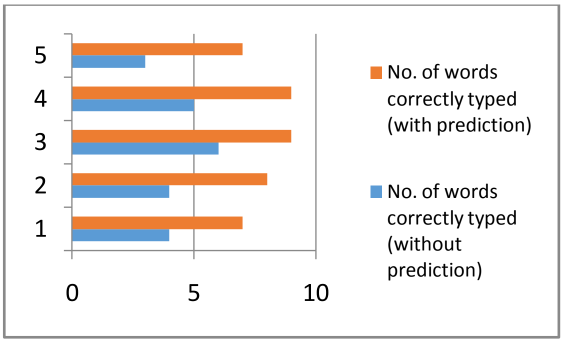

Figure 10: Comparison of the number of correct words typed with and without prediction

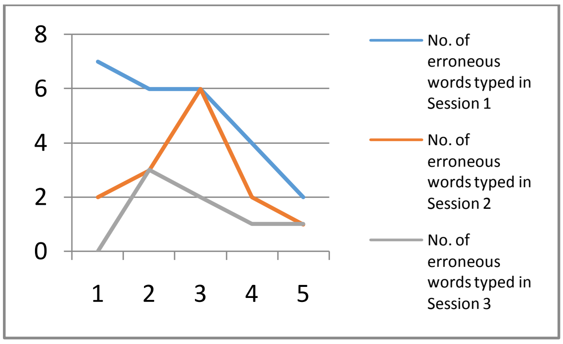

Figure 11: Comparison of the number of erroneous words typed over three sessions

The results of our evaluation concludes that See Through, if used with a comprehensive audio feedback and prediction system will prove to be highly successful after the users get accustomed to the system. Even though the number of keystrokes to key in a character might be slightly higher than the Multi Tap system, this system is much more effective than Multi Tap system in terms of number of error free words typed. It was also interesting to note that the users were interested to further use See Through rather than continue with the existing Multi Tap system.

Conclusion and Future Work

Even though mobile phones are a common tool today, they strongly rely on visual feedback for effective operation. At the outset, visually impaired users have trouble using them and this could result in frustration and social exclusion among some. Existing alternatives are not pragmatic and depend on rote memorization and this can lead to mistakes being made. We have developed an application which is a prototype of an innovative text-entry interface called 'See Through', which requires no memorization and no extra hardware and is inexpensive. It adds additional features to the design proposed by NavTap and attempts to enhance the same through the integration of intelligent systems. The results of our evaluation among target users show that the innovative design brings down the errors made by the users while typing. Our project caters to a section of the society which is not covered by existing mobile manufacturers. Since this is a software implementation and innovative design is the need of the day, it brings down costs radically and offers visually challenged people, a hope for the better.

Our innovation can be ever-improving. We plan to continue our research after we train the users and obtain feedback from them. Without commercial understanding, the research cannot proceed as the target users should decide the future scope of this project. We also plan to further enhance the parameters to be considered as we evaluate this system amongst the users. The keypad can be assigned different functionalities if the user wishes to. The keypad can also be extended to other languages as part of future work based on the advice of linguistic experts. After further study, we envision implementing an application for smart phones using the same principles, as an alternative to the various applications which are present today.

References

[1] Burrell, James, W, IV, 2001, “Method of Using a Nine Key Alphanumeric Binary Keyboard Combined with a Three Key Binary Control Keyboard”, Grant US6232892 B1.

[2] Nesbat, Saied B., 2003, “A System for Fast, Full-Text Entry for Small Electronic Devices”, Proc. 5th International Conference on Multimodal Interfaces, ACM, New York, NY, USA, pp. 4-11.

[3] Frey, Brian, Southern, Caleb, Romero, Mario, 2011, “Brailletouch: Mobile Texting for the Visually Impaired”, Proc. 6th International Conference on Universal Access in Human-Computer Interaction: Context Diversity – Vol. 3, Constantine Stephanidis (Ed.), Springer-Verlag, Berlin, Heidelberg, pp. 19-25.

[4] Lagoá, P., Santana, P., Guerreiro, T., Gonçalves, D., Jorge, J., 2007,“Blono: a New Mobile Text-entry Interface for the Visually Impaired”, Springer LNCS, Universal Access in HCI Part II, pp. 908– 917

[5] Gaba, Harish, Vinay and Desai, Dhaval and Desai, Archit and Pandey, Kumar, Sanjay, 2014, "Brailler: Enabling the Visually Impaired to Use Smartphones", International Journal of Current Engineering and Technology, Vol.4, No.5.

[6] Clawson, James and Lyons, Kent and Starner, Thad and Clarkson, Edward, 2005, "The Impacts of Limited Visual Feedback on Mobile Text Entry for the Twiddler and Mini-QWERTY Keyboards", Proc. Ninth IEEE International Symposium on Wearable Computers (ISWC '05), IEEE Computer Society, Washington, DC, USA, 170-177.

[7] Paisios, N., Rubinsteyn, A., and Subramanian, L., 2012, "Mobile brailler: Making touch-screen typing accessible to visually impaired users", Workshop on Frontiers in Accessibility for Pervasive Computing.

[8] Agreda, Victor, Jr., (2012, August 20),Fleksy is an alternate text input tool for the visually impaired [Blog post], Retrieved from http://www.tuaw.com/2012/08/20/fleksy-is-an-alternate-text-input-tool-for-the-visually-impaired/

[9] Tinwala, Hussain and Mackenzie, Scott, I., 2008, "Letterscroll: Text entry using a wheel for visually impaired users", Abstracts of the ACM Conference on Human Factors in Computing Systems, pp. 3153-3158.

[10] Guerreiro, Tiago, Lagoa, Paulo, Nicolau, Hugo, Santana, Pedro, Joaquim, Jorge, 2008, “Mobile Text-Entry Models for People with Disabilities”, Proc. 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction, Article No. 39, ACM New York, NY, USA

[11] Patra, Sabyasachi and Patel, Prathamesh and Velisetty, Karishma, 2014, "A Research on Android's Text to Speech Engine", International Journal of Students' Research in Technology and Management, Vol. 2, pp. 19-25.