A Flexible Keypad Reducing Keystrokes and Key Jamming for Cell Phones

Sabirat Rubya, Samiul Monir, Hasan Shahid Ferdous
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology
Email: sabiratbittamoni@gmail.com, samiulmonir@gmail.com, hasan.shahid@csebuet.org

***Abstract*—With the ever-increasing popularity of cell phone devices, text based services on such devices are becoming more and more popular. Problems with traditional keypads primarily lie with the placement of the letters alphabetically on the keys. This configuration is comparatively easy for the users to remember but can greatly limit the flexibility of finger movement, as well as require a higher number of keystrokes and key jamming. In attempting to resolve these issues, a novel innovative solution is proposed here, focusing on both the structure of human finger movements and ordering of letters on the keys based on their frequency of use. Simulations and performance measurement of our represented system have shown rapid reduction in key jamming by up to 57 percent, improvements in flexibility of finger movement by up to 11.5 percent and reduction in number of keystrokes by up to 34 percent.**

***Keywords*-Cell Phone, Finger Movement, Frequency of Alphabets, Human Factors, Keypad and Key Jamming.**

I. Introduction

Billions of people around the world use cellular phones in their everyday life for communications. These days, cell phones provide an incredible array of functions, and new applications are being developed at a rapid pace. With the improvement of modern technology, cell phones are no longer used for voice services only, but we can also take pictures and videos, send e-mails, chat on instant messengers, listen to music, get directions, and even watch television. Among these applications, text based services like SMS (Short Messaging Services) and emails are most common and widely used.

SMS is the common term for sending short (maximum of 160 characters including spaces) text messages using mobile phones. Text messaging has been a tremendous success in many countries, including Asian countries like Bangladesh, India, Singapore, and Malaysia. Malaysians, for example, were found to have sent 11.7 billion messages in the first three months of 2007, compared to only 7.4 billion in 2006 [1]. SMS is a quick, easy, and cheap way to communicate with anyone, anywhere, and at anytime.

To use SMS and many other functionalities of a cell phone, we need the help of a keypad for text entry. There exist a number of variations on these keypads such as Multi-tap Keypad, QWERTY Keypad, and Touch Screen Keypads. Multi-tap Keypads use a key for several alphabets and users may enter an alphabet by tapping keys a particular number of times. This keypad is the most commonly used one, as cell phones with this type of keypads are quite cheap. QWERTY and Touch Screen Keypads can have individual keys for both the alphabets as well as numerals. Often they are more expensive than the traditional ones.

Multi-tap keypads are widely used around the world. In this system the user presses the key multiple times to make a letter selection. For example, the key *2* is assigned with the letters *A*, *B*, and *C*, thus if a user wants to enter a *C*, then he/she has to press the key three times (222) successively as *C* is the third letter placed on the key. The process of typing becomes more complicated when the intended consecutive letters are placed on the same key. For example, to text the word *cab* the user must press the *2* key using the following pattern: 222 (pause) 2 (pause) 22. To select the correct letter on the key, the user must pause to determine the correct letter. This phenomenon is known as *Key Jamming*. Most of the mobile phones employ a time-out process in which the user waits for a specified time (typically one to two seconds) before attempting to enter the next letter, which is why multi-tap keypad is often criticized for being slow [2]. Also while designing this keypad, it was assumed that the probability of occurrence of all letters will be the same and hence, the letters are assigned to the keys in alphabetic order. As a consequence, a frequently occurring letter may remain in a key whose position is not flexible for fingers to reach and may require multiple key presses.

Considerable attention has been devoted to improve the traditional keypad system. For example, the flexibility of thumb movement while typing has been considered in [5] which also reduces key jamming to some extent. While designing this layout, the keys were ordered from anatomical point of view, according to thumbs' flexibility and pressure to reach them. Afterwards, the letters of the alphabet were arranged in the frequent keys based on the frequency of the letters. A different approach was followed in [4]. Their proposed architecture kept vowels mostly in the first position of the keys by taking into account the fact that probability of occurrence of vowels are higher than that of consonants. Thus, it could reduce total number of key pressing but no consideration was taken of key jamming or thumb movement. Another limitation was due to the placement of the letters on keys in their natural order. This ordering is easy to recognize and use but on the other hand, casts off the higher probability of occurrence of a consonant that appear latter in the alphabet than another one appearing before it. A practical example can be the higher chance of appearance of 'H' than 'G' or, 'N' than 'M'.

A notable motivation of our work came from the confinement of the discussed layouts as none of them have addressed together all the problems prevailing in traditional keypad. The algorithm that is described in this paper was inspired mainly to satisfy three design goals to increase the typing speed with mobile keypad. These include:

1) The flexibility of thumb movement should be increased.
2) Total number of key pressing should be decreased as much as possible.
3) The algorithm should be capable of considerable reduction of key jamming.

Following these three factors, a new keypad layout is designed where ordering and position of letters are different from traditional keypad but use of this keypad will reduce the total typing time as well as enhance the flexibility of finger movement.

The paper is organized as follows. Section II defines the performance criteria based on which we compared our proposed system with others. We elaborate the proposed system in Section III and compare it through extensive simulations in Section IV. Finally we conclude this work after stating our future research directions in Seciton V.

II. Performance Criteria

To understand and compare our proposed keypad, we may require having a clear understanding of what we mean by performance of a cellular keypad. We identify the following factors when we try to evaluate the performance of a keypad.

A. Number of Keystrokes:

A keystroke represents a key press action on the keypad, or other equivalent input device. The number of keystrokes is counted by the number of key presses while typing. For instance, to type 'MONKEY' with a traditional keypad, the total keystroke will be 13 (1 + 3 + 2 + 2 + 2 + 3). We will experiment and compare our proposed keypad with other keypads to see whether our proposed one can reduce keystrokes required to type a message.

B. Key Jamming:

When two consecutive letters to be written are in the same key of a keypad, then after writing the first letter one has to wait for a while to write the next. This phenomenon is called key jamming and this increases the total typing time required. For example, to type MONKEY, key jamming occurs twice (between the pairs M-O and O-N) in traditional keypad. One of our goals in this paper is to reduce key jamming while typing.

C. Flexibility of Thumb Movement:

This is the ease of moving the thumb to a particular key. A key is flexible when it is reachable by the finger (thumb) without much pressurizing the physical structure and the internal joints of the thumb. Sharmeen et al. [3] discussed about this property in details. They identified the two types of thumb movement - Flexion and Extension required to press a key and stated that pressure in the interphalangeal joint of a thumb increases with the decrease in the joint angles. So forward direction movements are convenient while lateral movements of thumb create extra stress on the user [6], [7], [8]. They showed that based on this principle, the order of the flexibility of the keys in a keypad is: $1>2>4>5>7>3>6>8>9$. We will consider this property while designing our proposed keypad.

III. The Proposed System

We have considered a one-handed and one-finger multi-tap keypad layout where all the typing will be done using only the thumb of the right hand. Typing blank spaces and capitalization of letters are not taken under consideration. While designing our keypad layout, we have kept 'space' in key '0', which is followed by most of the keypads. As occurrence of symbols and punctuation marks (**,** , **.** , **;** , @ , **!** , **?** , (,) etc) is much less frequent than the letters, we have assigned them to the most inconvenient key to press [3], i.e., key 9. The rest of the 8 numeric keys available are used to type the 26 alphabets. Hence, six keys of them will get three letters each and the other two keys will be mapped to four letters each.

To work with, first we classify the letters into three groups: Group-A, Group-B, and Group-C according to descending value of their frequency. In other words, the topmost 8 frequent letters are contained in Group-A, the next 8 higher frequency letters go to Group B and remaining 10 letters are assigned to Group-C. This classification is required to prevent a combination with two or more highly frequent letters to be assigned to the same key.

We know that the performance of any decision-making algorithm can be measured by the total cost or the total utility. An algorithm is considered to provide a better result if either total utility can be increased or total cost can be decreased. In our proposed algorithm, we have intended to increase the utility value. To make decisions like whether a particular combination of alphabets should be assigned to a key or not, we have given emphasis on assigning a good set of utility value according to the following factors:

1) Flexibility of a key according to movement of thumb.
2) Letter and combination of letters according to the probability of its occurrence:

The 26 letters of English alphabet and all possible two letter combinations generated from them need to have a utility value, which is explained later. For the time being, we assume that the utility of a single letter will be its frequency. In a similar way, the utility of a combination will be measured from the frequency of that combination.

The proposed algorithm basically works in a greedy approach. We aim to increase the total utility of the algorithm by assigning higher utility combinations to more convenient keys to reach. The utility of a three-letter combination assigned to a key is dependent on the following two factors:

1) The frequency or probability of occurrence of each letter in that combination.

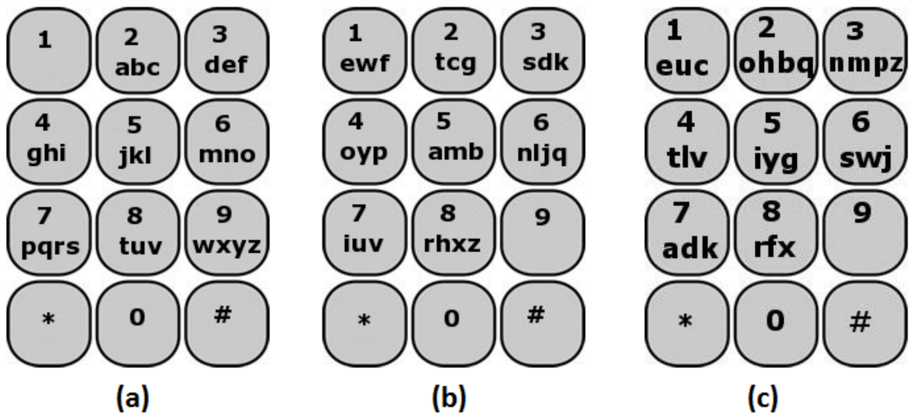

Figure 1. (a)Traditional Keypad (b)Proposed Keypad in [5] (c) Our Proposed Keypad.

2) The frequency or probability of occurrence of all possible unordered combinations of any two letters taken from that three-letter combination.

For example, the utility of the combination abc should be:

freq. of a	+	freq. of b	+	freq. of c
-freq. of ab	-	freq. of ba	-	freq. of ba
-freq. of cb	-	freq. of ca	-	freq. of ac

To understand why the frequency of all possible two-letter combinations are deducted, we need to consider that when two letters from a combination are consecutive in a word, key jamming will occur, thus increasing the wasted time. Therefore, the utility of a combination should decrease with increasing probability of consecutive occurrence of any two letters from the combination, as they would be contained in the same key. Another thing to notice is that, frequency of three-letter combinations does not contribute to the utility value as only pair-wise consecutive letters can cause key jamming.

Now we give insight to the detailed description of the proposed system. At first the utility of all the possible three-letter combinations are calculated using the method stated above. The sample database for this purpose has been selected from a web server where there is a list of SMS used by different people. We have only considered ordered combinations with no repeated letters, as a letter has to be assigned only once. So the combination sequence (abc, acd, ade, ... ,ayz, abd, abe,......,wxy,wxz,xyz) includes ${}^{26}C_3$ combinations. The combinations with higher utilities mean that the letters in those combinations are more frequent and jamming between them is less. So the only remaining concern is to assign the combinations with higher utilities to the more flexible keys so that the more frequent letters lie on the keys that are more flexible to use by the users. To do this, we sort the ${}^{26}C_3$ combinations according to non-ascending value of their utilities and the eight keys according to non-ascending value of their flexibilities. Then we proceed by placing the highest utility combination to the most convenient key, the second highest combination to the second most convenient key and so on until all eight keys are assigned to different combinations. We have to be very careful about a number of constraints at each step of assignment. While working with a particular three-letter combination, we need to be assured that the letters constituting the combination come from different classes (Group-A, B and C). This means no two letters from same group can be assigned to the same key. The logic behind this constraint is that if two letters from the same group are assigned to a key, one of them has to be in the second position in that key. So the total keystrokes will increase for the highly frequent letters in that key. In addition to this, when a combination is assigned to a key, we do not need to consider other combinations having alphabets from that particular combination.

The above process places 24 letters in 8 keys having three letters each. Remaining two letters are combined to any of the 8 keys by repeating the previous calculation of key jamming for four-letter combinations (the three letters already assigned to a key and another letter from those two letters) and thus finding suitable keys for these two letters. Finally, we have to order the letters of a combination in a proper way. For all the combinations assigned to the keys, we order them in non-ascending value of their frequencies. For instance, if the key 5 gets the combination abc and in our pre-calculation we find that, frequency of a, b, c are 20, 15 and 18 respectively, then we place abc in key 5 in the order acb. This will assure less keystroke for typing highly frequent letters. Our proposed algorithm can be summarized with a pseudo code in Fig. 2.

In this new layout, users will find the frequently used letters arranged in the flexible keys for the thumb to reach and require much less keystrokes. The waiting time between typing two consecutive letters will also be lessened as letters are assigned considering the key-jamming problem. Hence, this new keypad layout will be very much convenient and user friendly in comparison to the traditional keypad.

IV. Performance Studies

In this paper, we used text messages from a sample database obtained from [4]. It has 6300 words and about

```
Function generateKeypad(threeComb,allComb) returns keypad

Input: threeComb-a list of all possible three-letter combination.
       allComb-a list of all possible one and two-letter combination.

Output: keypad, a final keypad for our proposed system.

Local var:  utilitySet - a list of combined set of all possible three
letter combinations and their respective utility value.
            selectedComb - a particular combination.

Loop do
      pick a combination from threeComb
      calculate utility value for this combination
      add the combination and its utility value to utilitySet
end loop

Sort utilitySet according to non-ascending order

loop do until keypad full

      pick a combination from utilitySet
      if any letter of the combination is not exists in any of the
keypad combination and no other permutation is exists then

            add the combination to keypad
      end if
end loop
```

Figure 2. Algorithm for the proposed keypad.

34000 letters consisting of about 500 SMS messages. This training set was used to count the frequency of each character in alphabets and the frequency of all possible two- letter combinations. We used this information to find the most suitable keypad design according to our proposed system. Then we compared the performance of our proposed keypad with that of the traditional keypad and the keypad proposed by Azad et al. [5] according to the criteria discussed earlier. All simulations were coded using J2SE 1.6.

After arranging the keys according to our proposed algorithm using the sample database, the new keypad layout seemed like the one shown in Fig. 1(c). As the proposed layout is quite different from the traditional one, a learning time will be needed for the users to get used to this design. But as soon as they become experienced to type with this new keypad layout, they will be greatly benefitted.

As we have mentioned earlier, the prime motivation of our proposed algorithm was to reduce the total amount of keystrokes and key jamming as well as extend users flexibility. Different SMS databases with varied number of words were used for testing the performance based on total number of keystrokes, total amount of key jamming and number of key presses on both easily-reachable keys and inconvenient keys. By the word inconvenient, we imply the keys those need extension and movement to lateral direction of thumb to press as discussed earlier. Thus the keys 3, 7 and 8 are inconvenient while the other five keys are convenient or flexible. Testing came out with a promising result in comparison with the traditional keypad

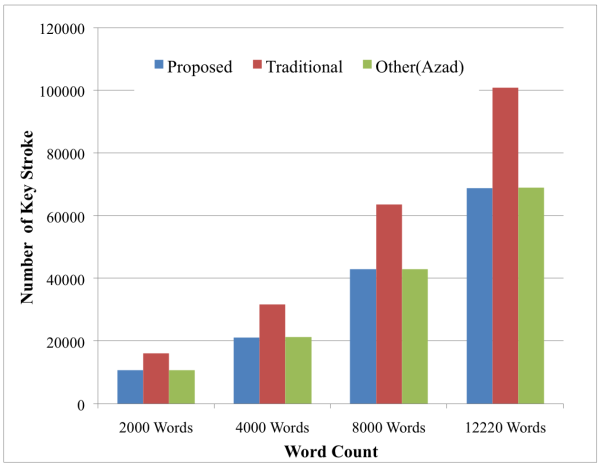

Figure 3. The Number of key strokes required.

Fig. 1(a) and Completely Enhanced Cell Phone Keypad proposed in [5].

Fig. 3 shows the performance of our proposed keypad in terms of the number of keystrokes required to type the words in our sample dataset. We can see that our proposed system decreases the keystrokes required by up to 34 percent than the standard keypad. Though the keypad proposed in [5] performs very good in terms of keystrokes, our proposed one outperform it by up to 0.5 percent.

Then we considered the number of key jamming wile typing Fig. 4. We can see that the standard keypad performs very low and the keypad proposed in [5], though performs better than the standard keypad, cannot reduce key jamming much. Our proposed keypad can decrease

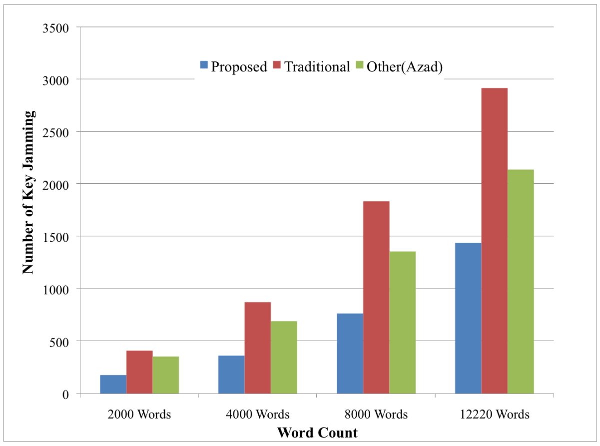

Figure 4. The number of key jamming occurred.

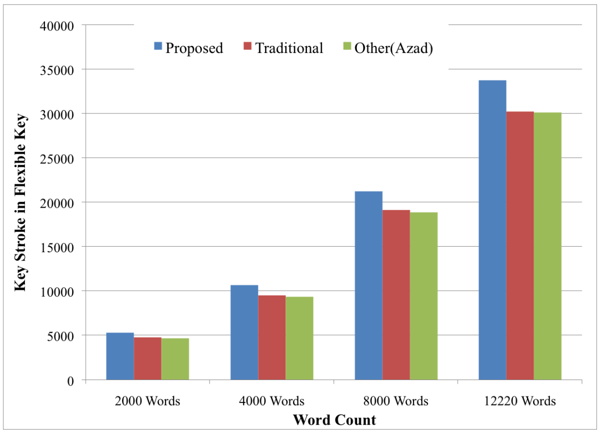

Figure 5. The number of flexible keystrokes.

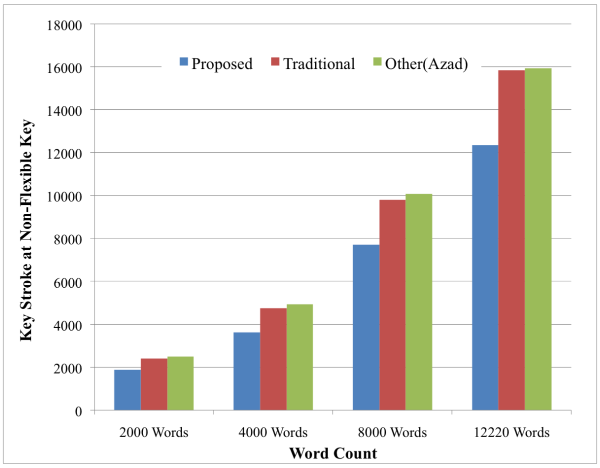

Figure 6. The number of inflexible keystrokes.

key jamming by up to 57 percent than the standard one and up to 50 percent than the one proposed by Azad et al. [5].

Finally we measured the performance of our proposed system in terms of key flexibility. We took two measures the number of flexible key presses and the number of inflexible key presses as identified earlier. From Fig. 5, we can see that our system improves the number of flexible keystrokes both than the traditional keypad and enhanced keypad [5]. From Fig. 6, we can also see that our system decreases the inflexible key press required by up to 25 percent and thus improves performance.

V. Future Works

Two factors can affect the designed layout of the keypad. First, we calculated frequency of alphabets from our sample database. Though our sample database is quite large, if changed, it can produce a slightly different result. Also, while assigning utility values to letters and combinations, one can consider other properties rather than frequency. Depending upon letters included in the groups and their utility values, the proposed algorithm can show some minor changes in its output. But the procedure remains same and can be generalized for other languages than English too. Since different languages have different number of alphabets, to accommodate those in fixed number of keys, number of letters included in a key will vary. Besides, some languages have joint characters in their words. To produce a model layout with less key pressing and key jamming and extended flexibility of finger movement for these languages, we can use the '*' or '#' key to implement other functionalities.

In future, we also plan to arrange user trials based on a prototype system on a mobile phone having our proposed layout to gain a practical impression of the typing speed. Moreover, the distance of finger movement in typing two consecutive letters can also be considered to see whether more performance improvements in terms of finger movement can be gained. In a nutshell, results shown here can be accumulated to give a thought for using the new layout with noticeable enhancement of typing experience as a replacement for the current text entry method.

VI. Conclusion

In this paper we have presented a novel approach for designing a multi-tap keypad for cellular phones considering frequency of alphabets, key jamming and flexibility of the thumb to use the keypad. Our new design has surpassed in all the performance criteria for text entry when compared to the standard and enhanced keypads. Though a learning period will be needed to get accustomed to the new layout, yet it will provide an accelerated speed for typing with a reduction of extra keystrokes and unnecessary key-jamming in the long run. We are now working to develop a theoretical work to find the optimality of our proposed algorithm. In that context, finding the closeness to optimality of our greedy heuristic algorithm remains as a major challenge.

REFERENCES

[1] The Star. (2007). 11.7 billion SMSes sent in first three months. Retrieved August 10, 2007, from “www.thestar.com.my/news/story.asp?file=/2007/8/8/nation /18525524&sec=nation”, as on October, 2011.

[2] S. I. Mackenzie .” Mobile Text Entry Using Three Keys”. *Proceedings of the Second Nordic Conference on Human-Computer Interaction*- NordiCHI 2002 (pp. 27-34). New York: ACM.

[3] R. Sharmeen, M. A. K. Azad, S. Ahmad and S. M. Kamruzzaman, ”Smart Bengali Cell Phone Keypad Layout”, in Proceedings of *Internation Conference on Computer and Information Technology*, (ICCIT 2005).

[4] S. N. Srirama, M. A. A. Faruque, M. A. S. Munni, “Altenative to Mobile Keypad Design: Improved Text Feed”, in *Journal of Computing Research Repository (CORR)*, vol. abs/1007.3, 2010.

[5] M. A. K. Azad, R. Sharmeen, S. Ahmad and S. M. Kamruzzaman, “Completely Enhanced Cell Phone Keypad”, in Proceedings of the *1st International Conference on Information Management and Business (IMB2005)*.

[6] Structure of human thumb: “www.anatomy-resources.com”, as on October, 2011.

[7] Structure of thumb joints: “www.handuniversity.com/topi cs.asp? Topic_ID=19”, as on October, 2011.

[8] Structure of thumb: “www.ncbi.nlm.nih.gov/entrez/”, as on October, 2011.