Mobile Phone Keypad Design for Fast Chinese Text Entry by Phonetic Spelling

by
Yuan-Ting Erica Huang
Department of Computer Science
University of British Columbia
yuanting@interchange.ubc.ca

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Bachelor of Science with Honours in Computer Science

The University of British Columbia
April 2006

ABSTRACT

The trend of using text messaging on mobile phones has grown rapidly in the last decade. However, mapping the alphabet to twelve phone keys introduces challenging ambiguities for text entry. This challenge is exacerbated in Chinese by the large phonetic alphabet and homophonic Chinese characters.

In response, we propose a novel algorithm to generate keypad layouts that reduce ambiguity for Chinese text entry. The layouts may greatly facilitate text messaging for Chinese users. Furthermore, the algorithm can also be applied to layout design for other ideographic languages (such as Korean and Japanese) and English T9 style entry.

The solution begins with a metric loosely based on Hick's Law to model and evaluate a keypad layout, estimating a user's performance time. Then a best-improvement, random-restart local search is used to generate high quality layouts according to the evaluation metric. Although we have not yet performed user studies, we outline a user study plan to compare the resulting layouts against successful commercial layouts.

In addition, there are some alternative ideas not implemented. In relevant sections, we will discuss alternatives to the design choices we made, some of which are promising for future work.

TABLE OF CONTENTS

INTRODUCTION

The growth of Short-Messages-Service (SMS) on mobile phones has been phenomenal in the last decade. The monthly volume has increased from 4 billion in 2000 to 24 billion in 2002 through GSM (Global System for Mobile communication) networks [1]. Due to the size limitation, text entry on a mobile phone has created interesting problems and become an active research area in Human Computer Interaction. Many text entry techniques and different interface designs have been developed to make text entry on mobile phones more efficient. However, the problem becomes much more difficult with Chinese text entry due to the large phonetic alphabet and homophonic Chinese characters. Still, there is a huge body of Chinese mobile phone users; the number has reached 320 million in 2004 [2]. Therefore, the Chinese text entry process represents a challenging and rewarding research problem.

In the rest of the introduction section, we provide some background information on how to input Chinese text and the reasons for focusing on the Zhu-yin phonetic spelling system. In the next section, we discuss the challenges posed by Chinese text entry with a phonetic spelling system. Then in the keypad evaluation section, we discuss the proposed evaluation metric and implementation as well as evaluation results of commercial layouts. In the keypad generation section, we discuss the algorithm developed to generate high-quality layouts, implementations and results. The application section discusses how our algorithm applies to other ideographic languages or even to English T9 style entry. We conclude by summarizing our contributions and discussing future work.

How to Input Chinese Text

Chinese is an ideographic language, and its minimal unit is a character, which can sometimes correspond to more than one pronunciation and more than one meaning. Due to the enormous character set, Chinese characters cannot be mapped to a keyboard for input into a computer system or an electronic device. Thus an intermediate step is necessary for Chinese text entry.

There are a variety of intermediate steps used for Chinese text entry, of which phonetic spelling is the most common one. Phonetic spelling is putting together letters to make a sound, as in English. Suppose a user wants to enter into a computer system the character “飛” (pronounced “fay”) which means the action “fly” in Chinese. He or she can spell out the sound “fay” using a special keyboard which has Chinese phonetic letters mapped onto it. The system then generates the list of characters pronounced “fay” for the user to select from, and the user will be able to choose the character ”飛” from the list.

There are two commonly used Chinese phonetic spelling systems, Pin-yin and Zhu-yin. The Pin-yin system uses the English alphabet and is widely used in China, while the Zhu-yin system has a separate alphabet and is used more in Taiwan and areas still using traditional Chinese.

Focus on the Zhu-Yin Phonetic Spelling System

Even though the Pin-yin system is more widely used for Chinese text entry, in this project, we focus on the Zhu-yin phonetic spelling system over Pin-yin for the following reasons:

1. Pin-yin shares the English alphabet, so designing solely for Chinese text entry is impractical in Pin-yin. Zhu-yin has its own alphabet.

2. The Zhu-yin system has a larger alphabet than English, and thus mapping letters on to keys becomes more difficult.

3. Current commercial systems using Zhu-yin all use an alphabetical order of some sort, but there is no standard Zhu-yin keyboard layout for a mobile phone. Thus, introducing a new order is feasible. Text entry with Pin-yin, which uses the English alphabet, adopts the standard keypad layout for English, and is thus hard to modify.

For the above reasons, using the Zhu-yin system as a case study for Chinese text entry efficiency is more practical, more interesting and more likely to be beneficial. However, the ideas proposed can be modified and applied to Pin-yin or phonetic spelling system for other languages.

Here is the Zhu-yin alphabet:

Consonants:

ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ

Vowels:

ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ

Both:

ㄧ	ㄨ	ㄩ

The Zhu-yin phonetic system consists of 37 letters and has certain rules. Please see Appendix A for the complete table with Pin-yin correspondence and the rules. To distinguish these letters from English letters, the Zhu-yin letters will be called *glyphs* in the rest of the paper.

THE PROBLEM

One challenge to Zhu-yin keypad layout design, which is also encountered in English keypad layout design, is that the number of glyphs is significantly greater than the number of keys available on a mobile phone. Thus, many different valid Zhu-yin sequences can be obtained by the same key sequence, so the user needs to take time to search for the desired Zhu-yin sequence.

Pervious work found that the *choice reaction time* – the time for a user to react and choose from a list multiple items – was the primary bottleneck in the Chinese text entry with keyboard [3]. (A more general term, *disambiguation time*, which includes the choice reaction time and also includes the time taken for a user to select operational keys, locate themselves on the screen etc., is also used in the rest of the paper.) This refers to selection among multiple homophonic Chinese characters.

Chinese text entry on a mobile phone introduces another stage that requires users to select from multiple items, that is to disambiguate between the Zhu-yin sequences generated by a key sequence. This is not a problem on full keyboards, since there are enough keys on the keyboard for 37 glyphs. However, the delay in human performance caused by choice reaction time is similar. Moreover, due to limited screen space for choices to be displayed at once, this delay becomes more critical in the text entry process.

Chinese text entry on a mobile phone also encounters the same problem with disambiguating between the characters corresponding to the sound represented by the Zhu-yin sequence. However, keypad layout does not affect the problem since the number of homophonic Chinese characters is fixed. Therefore, we focus on modeling the Zhu-yin sequence selection process.

The Zhu-yin Sequence Selection Process

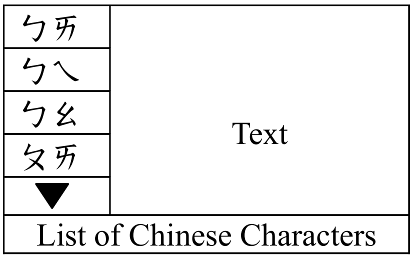
To be able to evaluate a keypad layout, we first model how a user selects a desired Zhu-yin sequence.

ㄅㄆㄇㄈ 1	ㄉㄊㄋㄌ 2	ㄍㄎㄏ 3
ㄐㄑㄒ 4	ㄓㄔㄕㄖ 5	ㄗㄘㄙ 6
ㄚㄛㄜㄝ 7	ㄞㄟㄠㄡ 8	ㄢㄣㄤ ㄥㄦ 9
	ㄧㄨㄩ 0	

Figure 1. Mobile Phone Keypad Layout used by Sony Ericsson

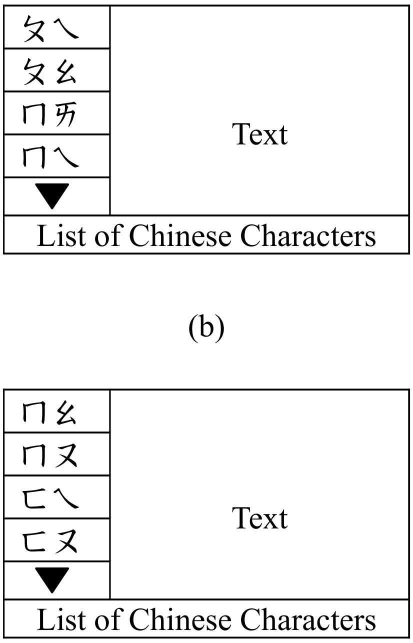
Suppose a user wants to enter the character “飛” into the mobile phone with the sample keypad layout in Figure 1. He or she will need to first key in 1 and then 8 to get the glyphs “ㄈ” and “ㄟ” for the sequence “ㄈㄟ”which makes the sound “fay.” The user then see the screen as shown in Figure 2(a) below. The layout of screen may vary for different mobile phone manufacturers, but the sample one is representative.

On the left is the list of four Zhu-yin sequences generated after the user presses the keys corresponding to the desired Zhu-yin glyphs. If the user cannot find the desired Zhu-yin sequence, he or she will select the down arrow to go to the next page and search again. In this case, the user does not see the desired sequence “ㄈㄟ” and selects the down arrow. The next screen, Figure 2(b), again does not display the desired sequence “ㄈㄟ” so the user selects the down arrow. Finally, as displayed in Figure 2(c), “ㄈㄟ” is the third on the list. The user now moves on to the list of characters, which is displayed at the bottom of the screen. After the user selects the desired character “飛” it will be displayed in the Text area on the right of the screen.



(a)

Figure 2. Screens displayed after the user enters keys 1 and 8



(b)

(c)

Figure 2. Screens displayed after the user enters keys 1 and 8

In the above example, entering the keys 1 and 8 generates 12 valid Zhu-yin sequences to select from. Figure 2 (a), (b) and (c) corresponds to the sequence of screen layouts that a user entering the character “飛” will see. In this example, a user has to go through all three sub-lists of Zhu-yin sequences, which complicates the selection process.

KEYPAD LAYOUT EVALUATION

Our goal is to reduce the disambiguation time to help speed up the text entry process. We first propose a keypad layout evaluation metric to investigate existing layouts based on the time for disambiguating among Zhu-yin sequences. After we are able to measure the efficiency of using a particular layout according to the disambiguation time, we can then develop an optimal keypad layout that minimizes this time.

Evaluation Metric

We developed a fitness function, denoted F(ZS, KL), using the model from Figure 2 as a performance measure for the selection of a particular Zhu-yin sequence given a keypad layout. ZS denotes the given Zhu-yin sequence, and KL denotes the given keypad layout. F(ZS, KL) estimates a cost proportional to the time taken to select one Zhu-yin sequence.

According to Hick's law [4], the time it takes for a user to choose from a number of items is not linear. If a user goes through the list of items one by one, the top choice will take the least time to select, and the last choice will take the most time. However, people subdivide the collection of choices rather than consider the choices sequentially; thus Hick's law has a logarithmic form. Based on this idea, the penalty for item selections is equally weighted. Every time the user selects a Zhu-yin sequence among the four in the list a cost of 10 is given. Every time the user selects the down arrow to see the next page a cost of 15 is given, to indicate that the time taken to look through all the choices and select next page is slightly longer. So using this evaluation scheme, selecting the Zhu-yin sequence "ㄈㄟ" scores a penalty 15 (first down arrow)+ 15 (second down arrow) + 10 (selection) = 40

The number of Zhu-yin sequences per screen, as mentioned before, may be different for different screen layouts, and the cost values 10 and 15 are somewhat arbitrary as well. However, the cost is not an absolute measure of the disambiguation time. It reflects the universal penalties for screen limitation and time taken for selection, and it should give a good measure of relative cost across different layouts.

Using the fitness function, we compute cost for the entry of a particular Zhu-yin sequence. The set of valid Zhu-yin sequences is finite, so we can then compute the overall cost for a particular keypad layout by summing up the individual costs. Since some Zhu-yin sequences occur more often than others, this sum should be weighted according to probability distribution of Zhu-yin sequences. Data for the probability distribution used in this project is from research by Tsai [5] using a large Chinese corpus provided by the Institute of Information Science, Academia Sinica in Taiwan. An SMS derived corpus might be more appropriate, but currently there is not a large enough corpus established for research.

Here is the proposed formula for overall cost:

$$C(KL) = \sum_{1}^{N} P_i * F(ZS_i, KL) \qquad (1)$$

ZS_i is the i^{th} Zhu-yin sequence and P_i is the probability of the i^{th} Zhu-yin sequence occurring. N is the total number of Zhu-yin sequences. In Chinese there are also 5 tones, which may be an element included in the Zhu-yin sequence. However, as tones usually are not labeled on commercial systems, we also did not consider Zhu-yin sequences with tones. In this project, we consider a total of 416 valid Zhu-yin sequences. Ignoring tones and inevitable rare variations, this is essentially an exhaustive list of all valid sequences.

Implementation

We have implemented a program for keypad evaluation in Java; it is set up in a way that we can fine tune or substitute the evaluation metric. The proposed formula implies that we go through each valid Zhu-yin sequence, convert it to its associated key sequence, compute the cost and then multiply by the associated frequency. However, we use a more efficient implementation, which is to go through each Zhu-yin sequence first, convert them to the corresponding key presses, and put the Zhu-yin sequences corresponding to the same key sequence, along with their associated probabilities, in to the same "key sequence bucket" so that computation of costs for them can be done together. This eliminates some redundant computations.

Alternatives

Alternative implementations might improve performance. One alternative implementation for storing the glyphs is to use a *Trie*, which is a tree structure that offers an efficient system for storing and accessing ordered data. It can probably make the algorithm more efficient; however, the current implementation works well, and using a Trie may make the implementation more difficult so is not implemented at the end.

Results

Here are some results from evaluating four commercial keypad layouts:

ㄅㄆㄇㄈ 1	ㄉㄊㄋㄌ 2	ㄍㄎㄏ 3
ㄐㄑㄒ 4	ㄓㄔㄕㄖ 5	ㄗㄘㄙ 6
ㄚㄛㄜㄝ 7	ㄞㄟㄠㄡ 8	ㄢㄣㄤ ㄥㄦ 9
	ㄧㄨㄩ 0	

Sony Ericsson (Score = 10,158,755)

ㄅㄉㄚ 1	ㄍㄐㄞ 2	ㄓㄗㄢㄦ 3
ㄆㄊㄛ 4	ㄎㄑㄟ 5	ㄔㄘㄣㄧ 6
ㄇㄋㄜ 7	ㄏㄒㄠㄡ 8	ㄕㄙㄤㄨ 9
ㄈㄌㄝ	0	ㄖㄥㄩ

Panasonic (Score = 2,136,900)

ㄅㄈㄋㄚ 1	ㄍㄐㄞㄧ 2	ㄓㄗㄢㄦ 3
ㄆㄉㄌㄛ 4	ㄎㄑㄟㄨ 5	ㄔㄘㄣㄤ 6
ㄇㄊㄜㄝ 7	ㄏㄒㄠ ㄡㄩ 8	ㄕㄖㄙㄥ 9
	0	

Motorola (Score = 2,769,955)

ㄅㄉㄚ 1	ㄍㄐㄗㄧ 2	ㄓㄞㄢ 3
ㄆㄊㄛ 4	ㄎㄑㄘㄨ 5	ㄔㄟㄣ 6
ㄇㄋㄜ 7	ㄏㄒㄙㄩ 8	ㄕㄠㄤ 9
ㄈㄌㄝ	0	ㄖㄡㄥㄦ

Okwap (Score = 2,266,000)

Figure 3. Evaluation of Four Commercial Layouts

One thing to notice is that, as mentioned before, there is no standard Zhu-Yin keyboard layout, and the number of keys used varies as well. The worst among the four commercial layouts we tested is from Sony Ericsson. Its layout uses 10 keys and follows Zhu-yin alphabetical order horizontally, which implies that all consonants are on keys 1 to 6, and most vowels on keys 7 to 0. Putting glyphs with similar properties (for example, glyphs that are all consonants or glyphs that are likely to occur in the same place in a Zhu-yin sequence) on the same key is likely to introduce ambiguity. On the other hand, the best layout of the four is from Panasonic, and the second best one is from Okwap, a Chinese brand for mobile phones. Both Panasonic and Okwap use 11 keys, and both roughly follow alphabetical order vertically and have a good mix of consonants and vowels on the same key. The Motorola keypad also follows the alphabetical order layout in the vertical fashion but only uses 9 keys so is at slight disadvantage.

KEYPAD GENERATION

Now we have defined, with the evaluation metric, what a good layout is. The goal is to develop an optimal keypad layout that minimizes cost according to our evaluation.

Breaking Alphabetical Order

As mentioned before, current commercial layouts follow alphabetical order roughly. However, prior studies on the effect of alphabetical layouts on keyboards have shown that an alphabetically organized layout is only slightly superior to a randomly organized one for novice users [6].

What happens if we apply the idea of breaking alphabetical order to mobile phone keypad design? Two keypad layouts are shown in Figure 4 below. Figure 4(a) is the sample layout in Figure 1, and Figure 4(b) is a similar layout with one difference – the glyph “ㄈ” is moved from key 1 to key 4.

ㄅㄆㄇㄈ 1	ㄉㄊㄋㄌ 2	ㄍㄎㄏ 3
ㄐㄑㄒ 4	ㄓㄔㄕㄖ 5	ㄗㄘㄙ 6
ㄚㄛㄜㄝ 7	ㄞㄟㄠㄡ 8	ㄢㄣㄤ ㄥㄦ 9
	ㄧㄨㄩ 0	

(a)

ㄅㄆㄇ 1	ㄉㄊㄋㄌ 2	ㄍㄎㄏ 3
ㄐㄑㄒㄈ 4	ㄓㄔㄕㄖ 5	ㄗㄘㄙ 6
ㄚㄛㄜㄝ 7	ㄞㄟㄠㄡ 8	ㄢㄣㄤ ㄥㄦ 9
	ㄧㄨㄩ 0	

(b)

Figure 4. Moving a Glyph on to a Different Key

As shown in the section on Zhu-yin sequence selection process, entering the keys 1 and 8 to obtain the sequence ㄈㄟ with the layout in Figure 4(a) results in 12 different valid sequences: ㄅㄞ, ㄅㄟ, ㄅㄠ, ㄆㄞ, ㄆㄟ, ㄆㄠ, ㄇㄞ, ㄇㄟ, ㄇㄠ, ㄇㄡ, ㄈㄟ and ㄈㄡ.

However, entering the keys 4 and 8 to obtain the sequence ㄈㄟ with the layout in Figure 4(b) only results in 2 different valid sequences: ㄈㄟ and ㄈㄡ. If we go through the Zhu-yin selection process of ㄈㄟ again using layout in Figure 4(b), with only 2 Zhu-yin sequences to choose from, it is much faster to enter the desired Zhu-yin sequence ㄈㄟ into the phone.

This, of course, is only an improvement for entering one particular Zhu-yin sequence, but we can see that improving a keypad layout by moving keys around is possible. In fact, according to our evaluation metric, the layout in Figure 4(a) scores 10,158,755, and the layout in Figure 4(b) scores 9,957,690. The change indeed makes an improvement. The goal now is essentially to minimize the number of Zhu-yin sequences generated by each key press sequence. In reality, this involves distributing all Zhu-yin sequences across the universe of key press sequences in a manner that respects the probability distribution of Zhu-yin sequences and our cost model.

However, the number of ways to map 37 glyphs to 12 keys is 12^{37}, so we cannot generate all possible layouts and evaluate them one by one to find the optimal layout. Therefore, we use a randomized stochastic search algorithm to search for a layout with minimum cost according to our evaluation metric.

Randomized Local Search

We use a combination of two standard search algorithms, best-improvement [7] and random restart, to search for high quality layouts according to our evaluation metric.

Our search space contains all possible Zhu-yin keypad layouts. For any given layout, we define a neighboring layout to be a layout with one difference in glyph placement. For example, the layouts in Figure 4(a) and Figure 4(b) are neighbors in our search space. Since there are 37 glyphs, and for each glyph there are 11 other keys it can be moved to, there are 37 * 11 = 407 neighbors for each keypad layout.

Our algorithm starts by randomly selecting a keypad layout. Then, using the best-improvement approach, it iteratively moves to the neighbor with the lowest cost according to the evaluation metric until it reaches a local minimum. This approach is not likely to find the global minimum, the optimal layout in the search space. However, random restarts locates the best of many local minima.

Implementation

The keypad generation program is implemented to incorporate the keypad evaluation program. First we implement the function to randomly assign glyphs to keys to create a randomly organized keyboard. Then we add the function to mutate a keyboard to its neighboring keyboard, by moving a glyph onto another key. The program starts with a randomly generated layout, creates all its neighbor layouts, and stores the one with best score. Then it repeats the process with the new layout. The function terminates when all the neighboring layouts score higher than the layout in focus. We can set how many trials we want to run the generation program, to get a variety of local minima.

Alternatives

We use the best-improvement approach because it is the most intuitive and easy to implement, but there are many other stochastic local search algorithms. We experimented with first-improvement instead of best-improvement. The first improvement approach picks the first layout with score lower than the layout we are comparing against. It diversifies the set of solutions when neighbors are considered in random order, while best-improvement always picks the same best neighbor to explore. The function for evaluating neighboring layouts can terminate once it finds one that improves the score, so it’s not necessary to create and evaluate all neighboring layouts. Moreover, it may be useful with a different definition of a “neighbor.” It takes a lot more steps to reach a local minimum, but the scores we get are similar to the ones result from the best-improvement approach. Other diversification techniques such as randomized iterative improvement may also worth exploring.

Results

ㄍㄒㄣㄦ 1	ㄋㄖㄜ 2	ㄅㄘㄞ 3
ㄈㄎㄢ 4	ㄇㄙㄝㄟ 5	ㄏㄐㄚㄩ 6
ㄌㄧ 7	ㄆㄓㄤ 8	ㄉㄠ 9
ㄑㄕㄥ	ㄔㄗㄛㄡ 0	ㄊㄨ

Score = 469,320

ㄒㄓㄣ 1	ㄆㄕㄥ 2	ㄉㄟㄡ 3
ㄊㄨ 4	ㄈㄍㄐㄢ 5	ㄎㄧㄝ 6
ㄑㄔㄘㄚ 7	ㄇㄙㄜ 8	ㄋㄖㄤ 9
ㄅㄗㄠ	ㄏㄩㄛ 0	ㄌㄞㄦ

Score = 481,280

Figure 5. Resulting Layouts with 12 Keys

Figure 5 shows the top 2 resulting layouts with number of keys = 12. Recall that the best score among the four commercial layouts, the Panasonic, scores 2,136,900 – about 4 times the best scoring layout generated by our algorithm. The Panasonic layout only uses 11 keys, so to be fair, we also run the algorithm with 11 keys. Here are the top 2 resulting layouts:

ㄍㄑㄚㄩ 1	ㄒㄓㄣ 2	ㄊㄨ 3
ㄇㄔㄙㄠ 4	ㄌㄢ 5	ㄋㄏㄖ ㄝㄞ 6
ㄅㄗㄥ 7	ㄆㄕㄟㄡ 8	ㄉㄛㄜㄦ 9
ㄎㄐㄧ	0	ㄈㄘㄤ

Score = 748,870

ㄑㄕㄞㄡ 1	ㄆㄈㄙ ㄝㄟ 2	ㄌㄧㄛ 3
ㄅㄗㄥ 4	ㄇㄔㄘㄠ 5	ㄉㄣㄦ 6
ㄎㄨ 7	ㄐㄓㄜ 8	ㄊㄖㄤ 9
ㄍㄒㄚㄩ	0	ㄋㄏㄢ

Score = 760,250

Figure 6. Resulting Layouts with 11 Keys

The best score is still much better than the existing commercial layouts.

The resulting layouts all have a fairly good mix of consonants and vowels as well as glyphs that are placed at different location in a sequence on the same key. One thing to notice is that some glyphs are tend to be grouped together, and some combinations of glyphs on the keys are quite similar, such as ㄇㄔㄙㄠ and ㄇㄔㄘㄠ, ㄅㄗㄠ and ㄅㄗㄥ, and ㄍㄑㄚㄩ and ㄍㄒㄚㄩ. This reflects the structure of the Zhu-yin system. Moreover, surprisingly, the glyphs are distributed quite evenly among keys. The cost function does not constraint the number of glyphs on a key; nevertheless, most keys have between 2 to 4 glyphs, and at most one per layout has 5 glyphs.

In addition, since there is not an established standard for number of keys on a mobile phone keypad for a Zhu-yin system, we ran an informal experiment of keypad generation with different number of keys for observation. The result of running a hundred trials with different number of keys is shown in Table 1 below:

Number of Keys	Maximum Score	Minimum Score
9	1,771,445	1,521,460
10	1,323,860	1,085,885
11	925,460	748,870
12	709,610	469,320
13	483,670	334,490
14	369,940	239,770
15	265,360	181,170

Table 1. Results Running Keypad Generation with Different Number of Keys

There are only 12 keys on a phone keypad, but trials with 13 to 15 keys are run so that we can observe the effect of increasing the number of keys. As expected, as the number of keys used increases, there will be less ambiguity in Zhu-yin sequence selection and thus a lower score. Surprisingly, there are big jumps in scores until the number of keys reaches 12, followed by reduced marginal gain. If 37 keys are used, all glyphs are mapped to different keys, and there will be no ambiguity and the cost should be minimal. Increasing from 9 keys to 10 keys and 10 keys to 11 keys result in great reductions in the cost; however, as the number of keys increases the benefit is less and less.

APPLICATIONS

As discussed before, we decided to focus on Chinese entry using the Zhu-yin over Pin-yin. However, the rules of forming a Pin-yin sequence are the same (see Appendix A for the correspondence between Zhu-yin and Pin-yin), so the algorithm proposed can easily be applied to the Pin-yin system. The algorithm can also apply to other ideographic languages using phonetic spelling, such as Japanese with the Hiragana spelling system and Korean with the Hangul system.

Moreover, the algorithm can apply to English text entry as well. T9 (which stands for Text on 9 keys) is a predictive text technology for mobile phones [8]. It's used widely for English text entry. T9 display the most frequent word with the corresponding key presses; however, if the word is not what the user looks for, the user needs to scan through the list of possible matches. This selection process is quite similar to Zhu-yin selection. The major difference is that with the Zhu-yin system, the user will always find the desired Zhu-yin sequence; however, with T9 the user can input an English word that is not in the dictionary. Nevertheless, the problem with choice reaction time is similar, and the algorithm proposed can easily be modified to accommodate the differences and be applied to English letter layouts.

However, as mentioned before, although our algorithm can apply easily to Pin-yin and English, we will run into the problem of an established alphabetical ordered layout.

CONCLUSIONS AND FUTURE WORKS

To study the efficiency problem of text entry with phonetic spelling, we used the Chinese text entry with the Zhu-yin system as a case study. We have identified the disambiguation time during the Zhu-yin sequence selection process as the primary bottleneck in the text entry process based on prior studies.

We have designed a metric which models the text entry process to evaluate cost for text entry with particular keypad layouts. Our contribution is that we have identified breaking alphabetical order as a possible solution to improve efficiency, and develop a novel algorithm that uses best improvement and random restart local search methods to generate specific good keypad layouts. Our algorithm has produced a set of high-quality layouts.

Conducting a user study to test real performance is the key upcoming task. We plan to compare users' performance on a high-quality layout from our algorithm against a commercial layout. In particular, we are interested in the efficiency and accuracy of user performance as well as user satisfaction. A detailed user study plan can be found in Appendix B.

Moreover, there are other factors that can affect the user performance. Currently the evaluation metric only takes into account the cognitive process time for making a selection. Another factor is the time for physical movement across keys, thus one evaluation criteria we can consider is to add in the physical movement distance between the keys.

In addition, even though the randomization of keypad layout has been shown to have small significance in terms of text entry efficiency, studies have shown that layouts with alphabetical tendency offer improvement to novice user performance [9]. Therefore, we can also consider the compromise of generating pseudo-alphabetical layouts, which make small changes to improve efficiency but retain some alphabetical ordering to help users learn the keypad layout.

$$C(KL,\lambda_1,\lambda_2) = \sum_{1}^{N} P_i * [\lambda_1 * F(ZS_i, KL) + \lambda_2 * H(ZS_i, KL) + (1 - \lambda_1 - \lambda_2) * K(ZS_i, KL)] \quad (2)$$

A proposed revised cost model that considers the additional two criteria is as shown in equation 2. $H(ZS_i, KL)$ is an estimated cost according to physical movement, and $K(ZS_i, KL)$ is a cost of derivation from a strict alphabetical layout. λ_1,λ_2 and $(1 - \lambda_1 - \lambda_2)$ are weights given to the three evaluation criteria. We leave the design of the two additional evaluation functions for future work.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. Steven A. Wolfman, for his guidance and support. I greatly appreciate his detailed feedbacks and the time he spent in the weekly meetings to discuss the research, which have helped tremendously towards the completion of the project. I am especially grateful for his constant encouragement and enthusiasm during all stages of the research.

REFERENCES

1. Global System for Mobile Communications. http://ww.gsmworld.com. Last retrieved: 14/04/2006.

2. China's mobile phone user reaches 320m. http://www.chinadaily.com.cn/english/doc/2004-11/23/content_393901.htm. Last retrieved: 14/04/2006.

3. Wang, J., Zhai, S. and Hui, Su. Chinese Input with Keyboard and Eye-Tracking – An Anatomical Study. Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI 2001, ACM, New York, NY, (2001), 349 – 356.

4. Hick, W. E. On the Rate of Gain of Information. Quarterly Journal of Experimental Psychology, (1952), 4:11-26.

5. Tsai, C. Similarities between Tonyong Pinyin and Hanyu Pinyin: Comparisons at the Syllable and Word Levels. http://research.chtsai.org/papers/pinyin-comparison.html. Last retrieved: 14/04/2006.

6. Norman, D. A. and Fisher, D., Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. Human Factors, 24(5), (1982), 509-519.

7. Hoos, H. and Stützle, T. Stochastic Local Search: Foundations and Applications, Morgan Kaufmann, San Francisco, USA, (2005), 263-71

8. T9 Text Input for Keypad Devices: http://www.tegic.com/. Last Retrieved: 19/04/2006

9. Zhai, S., Smith, B. A. Alphabetically Biased Virtual Keyboards Are Easier to User – Layout Does Matter. Extended Abstracts for Conference on Human Factors in Computing Systems on Human Factors in Computing Systems – CHI 2001, ACM, New York, NY, (2001)

APPENDIX A: COMPLETE ZHU-YIN TABLE AND RULES

Glyph	Pin-Yin	Position
ㄅ	b	I
ㄆ	p	I
ㄇ	m	I
ㄈ	f	I
ㄉ	d	I
ㄊ	t	I
ㄋ	n	I
ㄌ	l	I
ㄍ	g	I
ㄎ	k	I
ㄏ	h	I
ㄐ	j	I
ㄑ	q	I
ㄒ	x	I
ㄓ	zh	I,A
ㄔ	ch	I,A
ㄕ	sh	I,A
ㄖ	r	I,A
ㄗ	z	I,A
ㄘ	c	I,A
ㄙ	s	I,A

Glyph	Pin-Yin	Cat.
ㄚ	a	E,A
ㄛ	o	E,A
ㄜ	e	E,A
ㄝ	e	E,A
ㄞ	ai	E,A
ㄟ	ei	E,A
ㄠ	ao	E,A
ㄡ	ou	E,A
ㄢ	an	E,A
ㄣ	en	E,A
ㄤ	ang	E,A
ㄥ	eng	E,A
ㄦ	er	E,A
ㄧ	i / yi	I,M,A,E
ㄨ	u / wu	I,M,A,E
ㄩ	u / yu	I,M,A,E

I = initial
E = end
M = middle
A = alone

Rules:

There are 37 Zhu-yin letters (glyphs) in total. Each Zhu-yin sequence is made up of at most 3 glyphs to resemble the sound of a Chinese character. The column on the left are all consonants; the column on the right are mostly vowels, with the exception of ㄧ, ㄨ, and ㄩ which can act as both. There are four possible positions: a glyph can be the initial glyph to start the Zhu-yin sequence, be in the middle of the sequence of 3 glyphs, end the sequence or stand alone to make a “sequence” of one glyph.

Examples:

Zhu-yin sequence	Number of Glyphs	A Character Example
ㄣ	1	恩 (favor)
ㄈㄟ	2	飛 (fly)
ㄆㄧㄠ	3	飄 (float)

APPENDIX B: USER STUDY PLAN

Purpose

For the project "Mobile Phone Keypad Design for Fast Chinese Input by Phonetic Spelling," we propose to conduct a user study to test the best keypad layout according to our evaluation system. In particular, we would like (1) to examine how quickly and accurately people can enter Chinese text using with our best layout, in comparison to a dominant commercial layout and (2) to verify how novice users rate their comfort with our best layout, in comparison to a dominant commercial layout.

Users

The proposed number of test users for this experiment is twelve. All test users are required to be proficient in Chinese and experienced with Chinese text entry with Zhu-Yin phonetic system.

Experiment Description

To compare the efficiency and accuracy of text entry with the layouts, we will set up an environment on a desktop computer that simulates text entry on a mobile phone. We will provide Chinese text extracted from a corpus for the users to enter into the system. The users will perform the text entry tasks on both our best layout and a chosen commercial layout. The targeted users are required to be familiar with entering Chinese text with the Zhu-Yin phonetic spelling system.

Before the experiment there will be a training session for the user to become familiar with the text entry system. The training session will involve a fixed sequence of text snippets to enter, beginning with single characters and working up to phrases. This is to help the users to gradually get used to the system, so the text snippets are chosen and not selected at random. Then the users will have free time to continue practicing with a pool of pre-selected random snippets from the corpus. After the practice session, we will start the actual test session, in which we will collect data for analysis. Test snippets will be pre-selected at random from the corpus, subject to constraints on length and position within a document (e.g., at the start of sentences, full sentences, etc). All snippets will be consistent across subjects and across both layouts.

A single snippet session, both in the training and the test session, will proceed as follows:

(1) The user sees a message on the screen asking them to press the ENTER key when they are ready to proceed to the next text entry task. Instructions for the task also appear in the "instructions" portion of the screen where they will remain throughout the task.

(2) Once the user presses the key, the text to enter appears on the screen in Chinese. At this point, we begin timing for this exercise. The user now enters the text using the keypad layout currently illustrated on the numeric keypad of the computer. The process proceeds much as on a cell phone, with initial entry on the keypad followed by disambiguation of Zhu-Yin sequences, followed by disambiguation of Chinese characters. Each Zhu-Yin sequence and each character in the candidate list will have a numeric value assigned to it; the user can use the keys as normal numeric keys to make selections. As the user selects characters, they will appear immediately below the task text for verification. Consistently placed OK, CANCEL, and scrolling keys will be available to the users. If the user makes an entry error, they must hit CANCEL.

(3) The task ends either when the user completes it successfully (expected) or gives up by notifying the experimenters, who will advance to the next task. We will log all key presses, with times, during this process. The number of CANCEL presses is used to indicate the number of errors made.

We will inform users which session they are in during the experiment. At the end of both test sessions we will give out a brief survey, asking for users' feedback on comfort of usability on the layout, and at the end of the experiment we will give out a comparative survey. All surveys will be computer based.

The full session will be as follows, with estimated completion time for each session:

Introduction: 10 minutes
Layout #1 training session: 10 minutes
Layout #1 "free practice" session: 10 minutes
Layout #1 test session: 15 minutes
Layout #1 survey: 5 minutes
Layout #2 training session: 10 minutes

Layout #2 "free practice" session: 10 minutes
Layout #2 test session: 15 minutes
Layout #2 survey: 5 minutes
Comparative survey: 5 minutes

Total: 95 minutes

The order of layouts will be counterbalanced across subjects, and data will be compared within subjects.