GPK: An Efficient Special Symbol Input Method for Keyboards

Xiyuan He
Zhuohao Zhang*
Zhejiang University
Hangzhou, China
magnet@zju.edu.cn
zhuohaozhang@zju.edu.cn

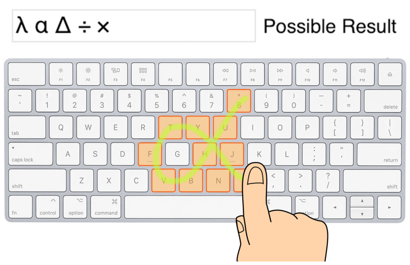

Figure 1: An overview of GPK. Users can glide on the keyboard and choose the special symbol they want to type

*Both authors contributed equally to this research

ABSTRACT

We introduce a novel typing technique for special symbols in the keyboard-only environment. The technique, called GPK (Gliding on Physical Keyboard), consists of two steps for entering special symbols. First, a user draws the special symbol on the keyboard by gliding over keys; Second, the user can select the desired symbol from the predicted symbols generated by GPK. Users can also switch from this mode to normal typing mode. We also present an application of this input technique based on web browsers. A user study with nine participants who are familiar with keyboard input showed the input efficiency of GPK. We compared the typing efficiency of GPK with other special symbol typing methods. We could deploy this method to office environments where users have desktop computers with a keyboard only. It could also inspire future work of integrating this method into word processors, document preparation systems and web environments.

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3308457

KEYWORDS
Input Methods; Special Symbol Input; Glide on Keyboards; Desktop Environment

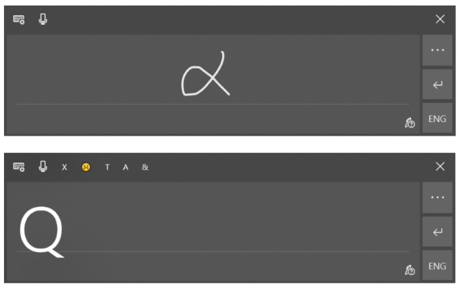

Figure 2: The virtual keyboard on Win10. Users can drag the mouse to draw characters, but it supports basic characters only (e.g. handwritten α was recognized as Q in the figure)

INTRODUCTION

Full-size keyboards have been the most common and efficient text input devices by far. However, it is still difficult for users to type in some special symbols that do not directly appear on the keyboard, like Greek letters or mathematical symbols. They are commonly used in theoretical or mathematical contexts. Currently, there are several existing and popular ways for special symbols input:

- Handwriting-based method: Use handwriting as an input method on some panels like touchable screens. Users are supposed to draw and select the right symbol predicted by the system (like in Figure 2);
- Spelling-based method: Use document preparation systems (e.g. LaTeX) or markup languages (e.g. Markdown) to spell the symbol, like '\alpha' for α. Alternatively, use search engines to search, copy and paste the symbol (e.g. search "Greek letters" and find the right one on the Internet);
- Selection-based method: Use a set of special symbols and choose from it, like "Advanced Symbols" function provided in Microsoft Word.

However, existing methods are still limited. For example, it is challenging and inefficient to use the handwriting-based method on desktop computers, as users can only drag mice to draw symbols in a virtual panel. Many special symbols are also difficult to spell. It is also an additional burden for users to acquire the right spellings of the symbols. To address the limitations, we envisioned a new method of special symbol input. To approximate this vision, we start by considering a scenario of drawing symbols on keyboards rather than dragging mice, which also saves the time of choosing from various symbols. We built GPK, a new method of gliding on the physical keyboard to input special symbols. Based on the input sequence and their time stamp, we designed an algorithm to recognize the symbol accurately. Then the algorithm gives a few possible symbols as feedback, and users can choose their expected result from them. For now, our algorithm reached an accuracy of 96.3% for recognizing the correct symbol in the top three predicted results. We designed a demo website which provides a tutorial of GPK and a training process, and we asked users to glide on the keyboard to select the right symbol. We then use the data as a data set around two thousand input sequences to calculate the accuracy. We only collected the data from users who agreed to our terms of data collecting.

In the following sections, we first review techniques of special symbol input methods. Next, we present our method and related algorithms. We then proposed a user study comparing different input methods, and report users' feedback to inform future work.

RELATED WORK

Special Symbol Input Method

As we introduced in the first part, there are three existing methods for typing special symbols.

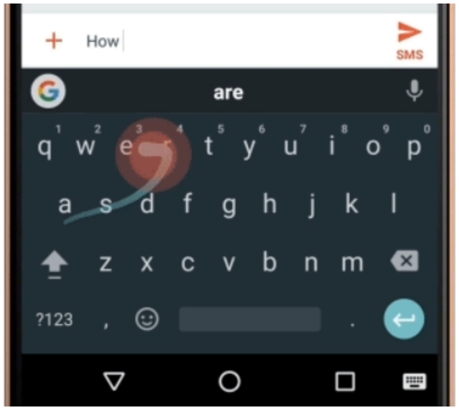

Figure 3: Gboard on Android is an example of sliding-to-type keyboard [3]

Figure 4: Swipe left by sliding on the physical keyboard of BlackBerry KEYone or KEY2 [4]

The first method of handwriting-based input usually appears on mobile devices like smartphones and tablets. Users can quickly draw the symbol on the virtual panel as input. However, this method usually requires a real screen for drawing. On desktop computers, it became a scenario of dragging mice to draw on a virtual panel on the screen.

Document preparation systems and markup languages usually provide math mode for users to input special symbols. Users can type in special symbols as mathematical symbols, like '\alpha' for α. Additionally, users also use spelling to look up special symbols in search engines and paste it in texts. However, in this case, users are required to master the spelling of every symbol, while they usually do not know all the right spellings.

The selection-based method is also popular for special symbol input. In popular word processors like Microsoft Word, users can select the special symbol from a series of symbols provided by the software. This method is accurate and easy to use, but it also takes much time for users to find the exact symbol from plenty of symbols. ,

Gliding on Keyboard

Several typing techniques of gliding on the keyboard have been proposed both in HCI literature and industries. Zhang et al. presented GestKeyboard[8], a technique that allows a user to perform gestures on physical keyboards. GestKeyboard uses Protractor[2], a novel gesture recognizer, to recognize gestures performed on a physical keyboard. However, this work focused on gestures input only with an entirely different method, while our work focuses on general symbols using a novel algorithm. GBoard(Google Board) is an industry-level application for gliding input[7]. The difference is that GPK enables gliding to draw the symbol's shape, while GBoard inputs a word by sliding through the letters. Another industry-level product for gliding on the keyboard would be BlackBerry KEYone and KEY2, which contains physical keyboards on Smartphones. Users can swipe left or right on the keyboard to perform gestures. Although these products have significantly improved the input efficiency of texts, how to type special symbols is still a problem. It is still mostly limited to the selection-based method to choose from a range of special symbols.

GLIDING ON PHYSICAL KEYBOARD

GPK is designed to fulfill the task of typing some special symbols in a straightforward and convenient way while other methods are not available or inefficient. We implemented an application of GPK on web browsers using JavaScript. Users can type in special symbols in two simple steps:

(1) First, users can glide on the keyboard wherever they want by his/her finger to draw the symbols. And they need not worry about pressing on keys inadvertently.
(2) Second, GPK will make a prediction based on the user's gliding to produce five possible results. Then users can use number key one to five to choose the right symbol they want to input.

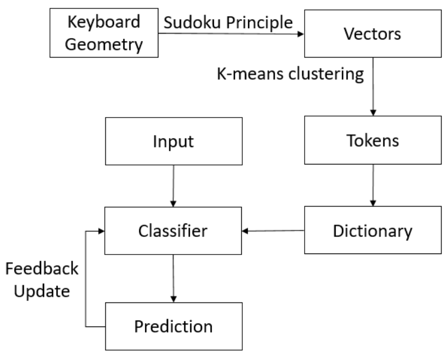

Figure 5: The main framework of the recognition

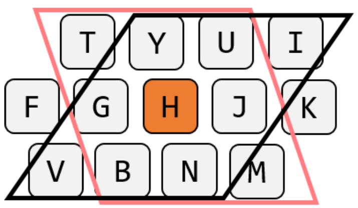

Figure 6: An example of Sudoku Principle: two Sudokus with Key 'H' being the center

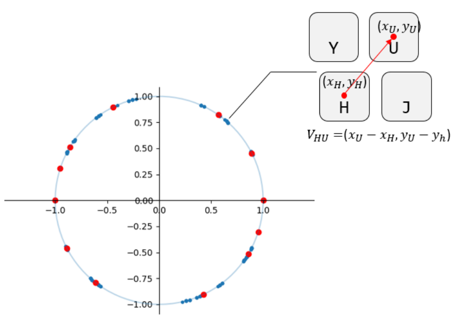

Figure 7: The clustering result. Blue dots are the vectors and red dots are the centroids (Tokens)

Also, users can switch from normal typing mode to GPK mode fluently. No other interactions are needed to intrigue the switch. Our algorithm will automatically recognize the input mode.

Implementation

Preparation. When given an input string, the main problem is how to extract the position information from the discrete typing sequence. In general, we made a connection to all the discrete keys so that they are in a temporal order. We then defined a *vector* as the location of the end key minus the start key. This vector is a tuple of x and y coordinates which points to one key from another. Then we introduced *Sudoku Principle*, which is to show the range of the next key to be pressed (As shown in Figure 6). We enumerated all the vectors that we can find based on *Sudoku Principle.* We then use K-means algorithm[1] to classify the vectors into 12 different groups and named them *tokens.* Figure 7 shows the clustering of the vectors. The red dots are the centroids. Then the clustered results(*tokens*) could be used in the prediction process.

Recognition. Figure 5 shows the main framework of our recognition process. Based on the preparation, given any input key sequence, we can transform the input into an array. Every element in the array, which is a number in the range of 0 to 11, represents a certain kind of token. We then created a dictionary for all the special symbols to be recognized based on the Hidden Markov Model(HMM)[6] and cognition by eight directions. Then we use the Dynamic Time Warping algorithm(DTW)[5] to calculate the similarity between the input and standard dictionary. Also, we train the DTW to make it more adaptable and accurate based on the large amount of user data that we have collected.

USER STUDY

In order to compare our method with other existing input methods, we conducted a two-session user study. We tested four methods including GPK among a group of users in keyboard input environment. We evaluate the input efficiency by measuring the input time of some certain sentences.

Methods

Participants. We conducted the study with 9 participants (six males, three females), with an average age of 30. None of the participants have experience with GPK before. All of the participants are familiar with typing on keyboards.

Procedure. The study contains two sessions, the first of which is a pilot session. We conducted a brief introduction of GPK to the participants until he/she understands its usage. Then participants are required to type in four sentences using four different input methods accordingly "as quickly and accurately as possible." The four methods are as followed:

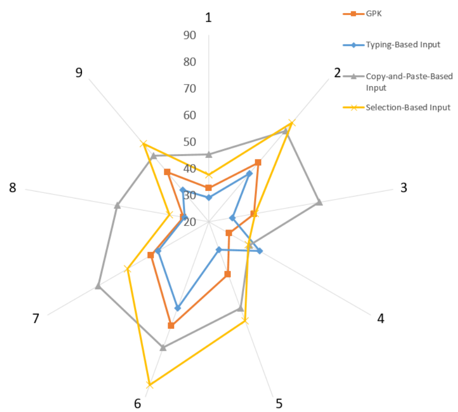

Figure 8: User study session 1: Typing-based method is slightly better than GPK with a brief introduction of this method

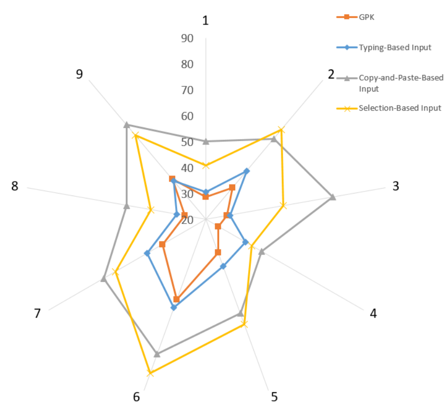

Figure 9: User study session 2: GPK became the most efficient method after a training process on the demo website

(1) GPK. Using the gliding input method as we designed. Users can glide and draw the special symbols and switch to normal input mode in order to type in normal contents;
(2) Typing-Based Input. Using a double dollar sign as the sign of special symbol input, and type in the symbol like "\alpha" for α. This method is generally used in LaTeX and Markdown;
(3) Selection-Based Input. Using Microsoft Word to type in. Participants can call out the panel of special symbols and select the symbol;
(4) Copy-and-Paste-Based Input. Participants can use the search engine to search for the right symbol by any means. Then they could copy and paste the symbol.

In the study, all the sentences for typing are about 20 words long (each for 98 characters), containing two or three special symbols. The sentences are copied randomly from common Wikipedia public pages. We believe they can represent and simulate users' daily input task. Also, a sentence of 20 words is long enough to obtain a stable result of typing time. Sample of sentences is shown as Sidebar 1.

In the second session, participants are required to take some training of GPK on our demo website. The website contains a thorough tutorial and training tasks for users to practice GPK. Participants could better master the skill of gliding on the keyboard and selecting symbols. Then, participants are required to retake the tests with the same procedure of the first session. The only difference is that the typing contents are renewed, with a length of around 20 words (98 characters exactly) for each.

During the two sessions, we recorded the duration time of users typing the sentences accurately for each method. The duration time could instruct the input efficiency of each method. Shorter time of typing using one method suggests that this method reaches a higher efficiency.

Result

Figure 8 and 9 are the results of our two-session user study. Each axis represents one participant in our study. The values on the axis represent the typing time for each method. We can see that in Figure 8, the first session, the result indicates that Method (2), typing-based method is the most efficient way to input special symbols, followed tightly by Method (1), GPK. The other two methods are much more inefficient according to the result. However, Method (2), typing-based-method, is only slightly better than Method (1), GPK.

In the second session, the result indicates that after a training process, Method (1), GPK, became the most efficient way of input special symbols. In the meantime, Method (2) is the second efficient method. The other two methods stay the same approximately.

DISCUSSION AND FUTURE WORK

The result clearly shows the effectiveness of GPK. In the first session, we gave the participants the right spellings of all the possible symbols, like \infty for ∞. However, in many scenarios, users do

Two of the samples in the user study:
Sample one: Alpha uppercase A, lowercase α is the first letter of the Greek alphabet. In the system of Greek numerals, it has a value of ω 1
Sample two: In mathematics, summations denoted with an enlarged capital Greek sigma symbol Σ σ is the addition of a sequence of λ numbers

Sidebar 1: Sample of sentences for typing: two or three special symbols within a 20-word sentence

not know the spelling of the special symbols. In this case, typing-based-method became the worst method, while GPK is still efficient as users knowing the shape of special symbols. The change in the second session also shows that after a training session, users can type in special symbols using GPK in a relatively high speed comparing to other methods.

Currently, GPK is implemented as an auxiliary application for typing, and it was also made to be available to users on various devices that have a physical keyboard by implementing it as a browser plugin. In the future, we are also aiming to expand GPK to be an additional built-in function of the physical keyboard or widely-used input method editor and allow user-defined symbols including Chinese characters, Russian alphabet and so on. Besides entering special symbols, GPK can also be deployed to support some system-level functions, like gliding a 'G' to open Google Web Search. ,

CONCLUSION

We have described a new approach to input special symbols called GPK, and its application for the keyboard-only environment. The two sessions of the user study we conducted demonstrate the efficiency of GPK. We could easily extend and transplant this approach into other software and environments, which could be a solution to special symbol input for a keyboard-only environment like clerks in offices.

ACKNOWLEDGEMENT

We want to thank Francis C.M. Lau, Yijie Wu for their mentorship and guidance, and the nine participants in our user study for their time of joining us.

REFERENCES

[1] E. Forgy. 1965. Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. *Biometrics* 21, 3 (1965), 768–769.
[2] Yang Li. 2010. Protractor: A Fast and Accurate Gesture Recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2169–2172. https://doi.org/10.1145/1753326.1753654
[3] Motorola. [n. d.]. Glide Type with Gboard - Moto Z Force (Droid). https://motorola-global-portal.custhelp.com/app/answers/indevice_detail/a_id/116397/p/30,6720,9833
[4] NieuweMobielNL. 2017. BlackBerry KEYone official product video. Video. Retrieved 2 January, 2019 from https://www.youtube.com/watch?v=2kv0hQF3J8c.
[5] Stan Salvador and Philip Chan. 2004. FastDTW: Toward accurate dynamic time warping in linear time and space. In *KDD workshop on mining temporal and sequential data*. Citeseer.
[6] R.L. Stratonovich. 1960. Conditional Markov Processes. *Teoriya Veroyatnostej i Ee Primeneniya* 5 (01 1960). https://doi.org/10.1137/1105015
[7] Wikipedia. 2019. Gboard — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Gboard
[8] Haimo Zhang and Yang Li. 2014. GestKeyboard: Enabling Gesture-based Interaction on Ordinary Physical Keyboard. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 195–196. https://doi.org/10.1145/2559206.2579485