UniWatch - Some Approaches Derived from UniGlyph to Allow Text Input on Tiny Devices Such as Connected Watches

Franck Poirier[1](✉) and Mohammed Belatar[2]

[1] Lab-STICC, University of Bretagne-Sud, Campus de Tohannic,
56000 Vannes, France
franck.poirier@univ-ubs.fr

[2] Technology and Telecom, Rue Dayat Aoua, 10090 Rabat, Morocco
mohammed.belatar@gmail.com

Abstract. Smartwatches are a fast-expanding type of interactive device that allow users to directly access to many applications on smartphone. At the moment, smartwatches lack a usable means of text entry. In this paper, we propose a new approach of text entry on smartwatches called UniWatch. At first, we give a state of the art concerning text entry on small devices. Then, we recall our past approaches of text entry and more particularly Uniglyph, a text input method for handheld devices that used a 4-button keyboard. Secondly, we describe and compare the different adaptations of Uniglyph for tiny connected devices such as smartwatches. All the proposed adaptations require only three buttons or three simple finger strokes on the screen. Thirdly, we examine the role of word completion and word prediction for such devices.

Keywords: Text input · Virtual keyboard · Word prediction · Connected watch · Smartwatch · Internet of things

1 Introduction

Smartwatches are on the wrist wearable computers that in addition to providing time give access to some functionalities of the smartphone. For example, the user can directly answer or make calls from his/her wrist, receive a notification of a message, take a snapshot or a short video… It is important to note that an essential functionality is absent from marketed smartwatches. Currently it is not possible to enter text directly on a virtual keyboard displayed on the watch screen.

The user must use the vocal assistant built-in application to communicate (S Voice on Android, Siri on IOS). The problem is that voice communication is not always possible or appropriate to the context of user interaction: voice is problematic in noisy environments and raise privacy issues in public spaces [19]. Voice communication is provided because there is currently no effective virtual keyboard on smartwatches.

From our point of view, text entry is a major functionality that should be present on all mobile or wearable devices [12]. The lack of usable text entry keyboard is probably a key reason for failure of wearable devices like smart glasses (i.e. Google Glass).

M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2015, LNCS 9170, pp. 554–562, 2015.
DOI: 10.1007/978-3-319-20916-6_51

In this paper, we propose some ways for text entry on smartwatches derived from UniGlyph. We call this approach UniWatch.

2 Related Works

For most of the smartphone applications, text entry is a key component. Smartwatches are usually presented as a new device that enables joint interaction with the smartphone. The paradox is that text entry is extremely limited on the smartwatch. We can therefore assume that the mass adoption of smartwatches is strongly conditioned by the possibility to enter short text with a smartwatch. That is why text entry on smartwatch is a major research challenge.

For the last two years, different text input methods for smartwatches have been proposed.

Few text entry methods are on the market for smartwatches, e.g. Fleksy [8], Minuum [13], Swype [21]. These three methods are based on a full QWERTY keyboard. Keys are so tiny that the finger touch does not hit the only desired key, the entry is disambiguated by lexical predictive algorithms. Predictive technologies are not perfect and are not suitable in typing abbreviations, acronyms, proper nouns… Due to the fat finger problem, it seems that a static QWERTY keyboard is not the right solution for smartwatches.

ZoomBoard [15] is one of the first methods based on a zooming user-interface (ZUI) paradigm. It provides a full QWERTY keyboard. The tiny keys around the finger press are iteratively enlarged, the user refines the finger position in order to point to the desired key, once this key is reached, zooming stops and the key is typed upon pressing.

Dunlop et al. [6] propose to divide the watch screen into seven zones, six big ambiguous keys, three at the top of the screen and three at the bottom and a center zone for the input entry field. OpenAdaptxt [14] is used for entry disambiguation and swipe gestures allow to change modes (alphabetical/numerical, lower/upper case, punctuation…), complete a word or enter a space.

DragKeys [4] is a circular keyboard composed of 8 ambigouous keys arranged around the text cursor. At most five letters are assigned to each key. To enter a letter, a first dragging gesture is made toward the key associated with the desired letter and a second dragging gesture in order to move the letter on the text cursor line.

Another approach is to use IR proximity sensors to capture gestures performed above the device, for example Gesture Watch [10] and HoverFlow [11]. This approach has the advantage to reduce screen occlusion but needs specific mechanisms, doesn't provide tactile feedback, and is not very discrete.

Xia et al. [23] use the watch face as a multi-degree-of-freedom mechanical interface. Their proof of concept supports continuous 2D panning on the watch screen, twist, tilt and click. They developed a series of example applications but don't envisage to input text with this kind of approach.

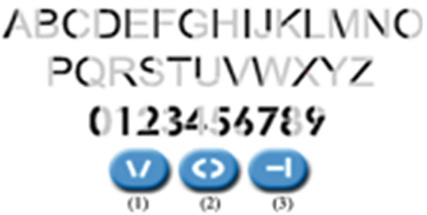

Fig. 1. The UniGlyph character set and the associated input keys: diagonal-shape key, loop-shape key, straight-shape key.

3 Uniglyph Text Input Method

UniGlyph [17] is a text entry method for handheld devices derived from Glyph [16, 18]. The methods of the Glyph family are based on the structure of Latin characters composed by a specific sequence of primitive shapes (curve, stroke, loop…).

For UniGlyph, the set of primitive shapes is reduced to only 3 symbols: (1) diagonal stroke, (2) curve and (3) horizontal or vertical line. Each primitive shape is dedicated to one key of the keypad called respectively diagonal-shape key, loop-shape key and straight-shape key.

Each letter of the English alphabet is represented by only one primitive shape according to the shape of the uppercase letter. In order to recall the coded key, the user needs to follow a very simple rule (Fig. 1):

- if the capital letter contains a diagonal stroke, then click on the diagonal-shape key (1);
- otherwise, if it contains a loop or a curving stroke, then click on the loop-shape key (2);
- otherwise, click on the straight-shape key (3).

As there are many more characters than primitives, each primitive corresponds to a set of letters. The expected word is deduced by a linguistic predictor like for all the ambiguous keyboards (T9®, SureType®, iTap®…).

The UniGlyph keypad contains three shape keys and one command key used to jump to the different input modes and to select the expected word.

4 Initial Design of UniWatch - An Adaptation of UniGlyph for Tiny Connected Devices

Form Factor Problem. The smartwatch screen is very small, from 1.2 (Pebble) to 1.6 inch. (Galaxy Gear). So, it is impossible to finger tap on a complete keyboard on such a so tiny screen in order to enter text. A smartwatch screen can just contain a small number of keys, buttons or icons. The solution proposed by Dunlop and al. [6] with seven keys on the screen occupies the whole screen, without referring to the magic number of Miller, it seems reasonable to have fewer keys on the watch screen. It is

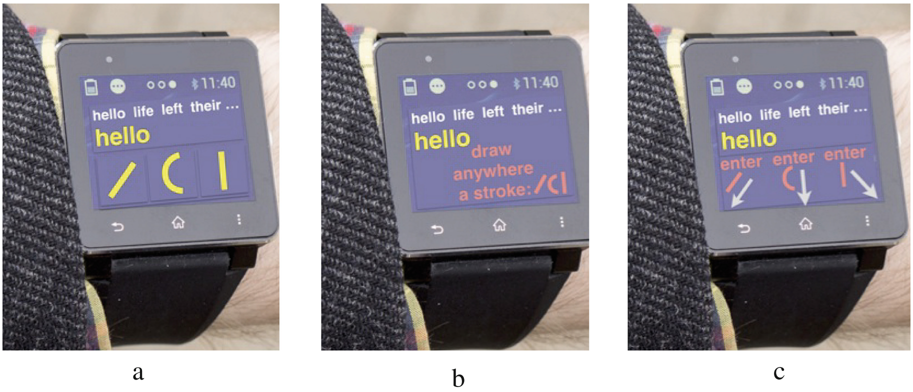

Fig. 2. Three approaches for texting: a- one key per input primitive shape, b- directly drawing the stroke (‘/’, ‘(’, ‘|’), c- one flick per input primitive shape. (Red text and arrows are only for explaining the figure. They are not displayed on the screen of the watch) (Colour figure online).

reasonable to enter text on a small keyboard only if the keyboard contains just very few buttons.

The UniGlyph approach is a good candidate for text entry on smartwatches because it minimizes the keypad to only 4 keys, even 3 if the commands are entered by a gesture directly applied on the watch. Gesture Watch [10], HoverFlow [11] or Xia and al. [23] have shown that sensor-based gestures (opposed to touch-based gestures) are suitable for controlling the text entry. More generally, with small devices or in mobility, it is better to combine the strengths of multi-touch gesture with motion- sensing gestures [9].

UniWatch, the adapted version of UniGlyph requires only three keys. It is especially well suited to input text on a smartwatch.

We propose different ways for entering text on the smartwatch screen with UniWatch.

Ambiguous Key Approach. The direct adaptation of UniGlyph is to use a 3-key keypad, each key corresponds to one input primitive. The original command-key can be replaced by sensor-based or touch-based gesture. The easiest solution is to directly touch the text field in order to validate one of the predicted words. These three keys can be placed on the lower side of the screen (Fig. 2-a) or in each corner. The interaction technique consists in button taping.

As with the method proposed by Dunlop [6] each key is ambiguous, a disambiguation engine gives word completion and word prediction. Another common feature is the number of keys across the width of the screen (3). Considering the finger size and the screen size, it seems difficult to put more than three or four keys at the bottom of the screen. A simple calculation shows that these keys occupy around 20 % to 25 % of the screen space.

With this approach, the user interaction is limited to single taps on the keys. Due to the size of the keys, even on the go, the risk of error is very low.

Single-Stroke Entry Approach. Another way is to use the touch screen capability by directly drawing the shape of the input primitives (diagonal stroke, curve or straight line) on the screen (Fig. 2-b). This approach is not new if we refer to the watch AT-550 made by Casio in 1984 [3]. With this watch, the user entered a calculation by drawing on the watch screen with a stylus, one after another, each operand. In 2014, this approach is re-used by Microsoft Research in the Analog Keyboard Project [2].

In our case, according to UniGlyph method, the primitive shapes are reduced to only 3 symbols (‘/’, ‘(’, ‘|’), each one corresponds to a single stroke and each one is quite different from the others. Finger drawing on the screen watch is easy and comfortable on a so tiny screen. In this way, the risk of error is also very low.

The advantage compared to the previous approach is that screen is not occupied by buttons.

Flick Gesture Entry Approach. The third approach is based on flick gestures on the screen to enter the input primitives and to control all the process steps of text entry. Flick gestures are used for a long time [22]. A flick gesture is a particularly fast way for entering a command, the gesture direction is significant but not the amplitude.

The flick gestures can be executed from the center towards one side of the screen (top, bottom, left or right side).

Figure 2-c shows one of the possible mappings, a flick bottom down-left is for the diagonal stroke (‘/’), a flick down is for the curve stroke (‘(’) and a flick down-right for the straight stroke (‘|’).

The difference with the previous approach is that the mapping between the flick and the primitive shape is arbitrary, the user must learn it. However, as there are only three primitive shapes, the mapping is very easy to learn and remember. With the *single-stroke entry* approach, the user directly writes the primitive shape on the screen. He/she just thinks which shape is associated to the desired letter. With the *flick gesture entry* approach, the user must think to the right key and, in addition, to the mapping between the key and the flick gesture.

5 Qualitative Comparison of Approaches

Each approach has some advantages and disadvantages. Text typing involves complex and numerous motor, perceptual and cognitive processes [5]. Because of this complexity, it is impossible to decide which is the best approach. Decision criteria must be established and the right approach must be chosen according to these criteria.

Table 1 presents a comparison of the three proposed approaches. Some drawbacks are associated with the user and other are related to the characteristics and limitations of the watch.

The decision taken is to go with the solution that favors the usability which means the ease of use, the familiarity, the comfort, the reliability and the entry speed.

The advantages and disadvantages are interpreted according to the concept of usability. Table 2 presents a subjective evaluation of the three methods in regard to the criteria of ease of use, familiarity, comfort, reliability and entry speed.

The *ambiguous key* approach gets the best score (12 points), the *flick gesture entry* approach follows (9 points) and the *single-stroke entry* approach seems to be the less usable (8 points).

Table 1. Comparison of the three approaches

Ambiguous keys	Single-stroke entry	Flick gesture entry
Advantages	Advantages	Advantages
Easy to use without learning (labeled keys).	Familiar mode of interaction (handwriting shapes).	Quickness of the flick gesture.
Classical button-based interaction.	Easy to draw primitive shapes.	Practically no risk of gesture recognition error.
Error-proof solution (button press).	Direct mapping between shape handwriting and desired letter.	
	Buttonless interaction.	
Disadvantages	Disadvantages	Disadvantages
Screen space occupation.	Lack of guidance on the screen.	Lack of mapping between flick orientation and desired letter.
Poor exploitation of tactile interaction possibilities (only taps on button).	Handwriting feedback superimposed on the information displayed.	Need to recall two informations: the right coding primitive shape and the associated flick gesture.
	Occlusion due to finger drawing.	
	Small risk of handwriting recognition error.	
	Slowness of handwriting process.	

Table 2. Evaluation according to the criteria of usability

Score range from 1 (low) to 3 (high)	Ambiguous keys	Single-stroke entry	Flick gesture entry
Ease of use	3	2	1
Familiarity	2	2	1
Comfort	2	1	2
Reliability	3	2	2
Entry speed	2	1	3
Total score	12	8	9

6 Quantitative Comparison of Approaches

It would also be interesting to base the comparison of the three approaches on a quantitative prediction model. The most suitable model is KLM (Keystroke Level Model). KLM is part of the wider GOMS-related work of Card, Moran, and Newell based on the Model Human Processor (MHP) proposed by the same authors. KLM is used to estimate the time taken to complete simple data input tasks by combining few input operators associated to timing constants. The main advantage of KLM is to describe tasks as a sequence of the operators and predict user interaction times without needing to create prototypes.

El Batran and Dunlop [7] have extended KLM for mobile touch interaction with three new operators for three new interaction techniques on mobile devices: tap, swipe and zoom. The extended KLM predicts user movement times for swiping (MT_S), taping (MT_T) and zooming (MT_Z),.

Based on this model, we can estimate the time for entering a n-length word (T_W) with our three approaches:

- $Tw = T_M + MT_T^n + T_M + MT_T$ for the *ambiguous key* approach
- $Tw = T_M + MT_D^n + T_M + MT_T$ for the *single – stroke entry* approach
- $Tw = T_M + MT_S^n + T_M + MT_T$ for the *flick gesture entry* approach

where T_M is the time for mental preparation, MT_T is the movement time for taping a button, MT_D is the movement time for drawing a primitive shape and MT_S is the movement time for flicking (flicking and swiping are considered equivalent).

In each expression, the first term (T_M) corresponds to the time spent for mentally preparing the touch operators, the second term (MT^n) is the time for entering a n-length word, the third time (T_M) is needed for scanning the word prediction list and the last term (MT) is the time for choosing the desired word.

According to Fitts' law, MT is expressed as:

$$MT = a + B \,.\, ID \text{ and } ID = \log_2(D/W + 1)$$

where ID is the index of difficulty, W is the width of the target and D is the amplitude of the movement. The coefficients a and b are usually determined empirically for a given device (mouse pointing, finger pointing, stylus pointing…).

In our case, for *button entry* or *flick gesture entry*, W and D are between one third and one half of the screen size, the D/W ratio is in the order of 1. Consequently, the index of difficulty is approximately 1. For this value of ID, El Batran and Dunlop found a predicted time for flicking of 70 ms and a predicted time for button pointing of 80 ms.

For our part, we experimentally found that MT_D, the time for finger drawing a primitive shape on the watch screen is approximately 100 ms.

In conclusion, there is no significant difference between our approaches from the total interaction time perspective (T_W).

7 The Problem of Word Prediction

Whatever the chosen approach, the text input is ambiguous. The expected word must be deduced by a linguistic predictor then validated by the user. In the context of tiny devices used on the go, the lexical prediction is absolutely essential in order to facilitate text entry.

The simplest way is to use the default predictor of the smartphone linked to the watch (QuickType for IOS, Next Word Prediction for Android, Sense Input for HTC…).

Assuming that only very simple posts with abbreviations such as TTYL (*Talk To You Later*), IDTS (*I Don't Think So*), RYOK (*Are You OK*), SYT (*See You Later*), IDK (*I Don't Know*), B4 (*Before*), TY (*Thank You*)… will be entered on the tactile screen of a connected watch, it must be preferable to use a specific predictor [1].

Moreover as the typing sentences are very short and probably not syntactically correct, the default linguistic predictor is not suitable. The most useful and most effective is just to present the word completion and the current word prediction without taking account of the all context of the sentence.

In order to personalize and speed up text entry, a limited set of predefined sentences should be fixed by the watch user. For example, "*I can't answer you, I'm doing my jogging*" or "*at what time you will go back home*".

It is important to note that processes that allow to speed up the entry, such as keyboard shortcuts, access keys, hot keys or word completion are especially appreciated by users. They allow to reduce the number of interaction and to increase the pace of interaction. They correspond to one of the 8 Golden Rules expressed by Shneiderman ("*Enable Frequent Users to Use Shortcuts")* [20] and also one of the eight ISO-Standard 9241-110 Dialogue Principles ("*Suitability for Individualization*").

The fact is that a simple sentence consists of around 8 to 12 words. With software keyboards for mobile devices or even smart watches, a trained user achieves an average of 8 to 12 words per minute (wpm). So, without shortcuts, writing a simple phrase takes more than one minute. In the specific context of interaction with a smart watch, one minute or more for texting is too long. As it is difficult to memorize a long list of abbreviations, a list of 25 to at most 40 common English abbreviations seems to be sufficient for covering lots of situations and speeding up the text entry.

8 Discussion and Future Directions

In this paper, we have analyzed different recent works on smartwatch text entry. We have designed a new approach called UniWatch derived from the UniGlyph method and have explored three input strategies based on touch buttons, finger drawing and flick gestures. We have performed a qualitative and quantitative comparison of these approaches. We have found that there is no significant difference between them from the quantitative point of view. On the other hand, from the qualitative point of view, the *ambiguous key* approach based on key taps has been judged more usable. Whatever the approaches, we have insisted on the necessity of a specific word predictor well fitted for short and not syntactically correct sentences and the importance of textual shortcuts.

In conclusion, taking into account the qualitative and quantitative analyzes, we think that the *ambiguous key* approach is preferable because it implies a better feedback (the primitive shape is recalled on the key), it is easiest to use (minimization of the working memory load compared to the *flick gesture entry* approach), it is quick and reliable.

In the next future, we will develop a proof of concept prototype of UniWatch based on ambiguous keys and textual shortcuts adapted to the context of use of a smartwatch.

References

1. Agarwal, S., Arora, S.: Context based word prediction for texting language. In: Proceedings of the 8th RIAO International Conference on Large-Scale Semantic Access to Content (2007)
2. The Analog Keyboard Project. Microsoft Research (2014). http://research.microsoft.com/en-us/um/redmond/projects/analogkeyboard

3. Buxton, B. The Mad Dash Toward Touch Technology. Business Week, 21 October 2009
4. Cho, H., Kim, M., Seo, K.: A text technique for wrist-worn watches with tiny touchscreens. In: Proceedings of ACM UIST 2014, pp. 79−80. ACM Press, Honolulu (2014)
5. Cooper, W.E.: Cognitive Aspects of Skilled Typewriting. Springer-Verlag, New York (1983)
6. Dunlop, M., Komninos, A., Durga, N.: Towards high quality text entry on smartwatches. In: Proceedings of ACM CHI 2014, pp. 2365−2370. ACM Press, Totonto (2014)
7. El Batran, K., Dunlop, M.: Enhancing KLM (keystroke-level model) to fit touch screen mobile devices. In: Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 283−286. ACM Press (2014)
8. Fleksy keyboard. www.fleksy.com
9. Hinckley, K, Song, H.: Sensor synaesthesia: touch in motion, and motion in touch. In: Proceedings of ACM CHI 2011, pp. 801−810. ACM Press, Vancouver (2011)
10. Kim, J., He, J., Lyons, K., Starner, T.: The gesture watch: a wireless contact-free gesture based wrist interface. In: Proceedings of 6th International Semantic Web Conference ISWC 2007, pp. 11−13, Busan (2007)
11. Kratz, S., Rohs, M.: Hoverflow: exploring around-device interaction with IR distance sensors. In: Proceedings of MobileHCI 2009, pp. 1−4. ACM Press, Bonn (2009)
12. MacKenzie, S., Tanaka-Ishii, K.: Text Entry Systems: Mobility, Accessibility, Universality. Morgan Kaufmann Publishers, San Francisco (2007)
13. Minuum keyboard. www.minuum.com
14. Montaparti, S., Dona, P., Durga, N. Meo, R. D.: OpenAdaptxt: an open source enabling technology for high quality text entry. In: Proceedings of CHI Workshop on Designing & Evaluating Text Entry. ACM Press (2012)
15. Oney, S., Harrison, C., Ogan, A., Wiese, J.: ZoomBoard: A Diminutive QWERTY Soft Keyboard Using Iterative Zooming for Ultra-Small Devices. In: Proceedings of ACM CHI 2013, pp. 2799−2802. ACM Press. Paris (2013)
16. Poirier, F.: Glyph: a new stroke-alphabet for stylus-based or key-based text entry. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) Proceedings of HCI International 2005. Springer, Heidelberg (2005)
17. Poirier, F., Belatar, M.: UniGlyph: only one keystroke per character on a 4-button minimal keypad for key-based text entry. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S. D.J. (eds.) Proceedings of HCI International 2007, vol. 2007, pp. 479–483. Springer, Heidelberg (2007)
18. Poirier, F.: Text entry methods for handheld devices or for AAC writing system. In: Proceedings of ACM CHI 2012. ACM Press, Austin(2012)
19. Sawhney, N., Schmandt, C.: Nomadicradio: speechand audio interaction for contextual messaging in nomadic environment. ACM Trans. Comput. Hum. Interact. **7**(3), 353–383 (2000)
20. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison Wesley, Boston (2004)
21. Swype keyboard. www.swype.com
22. Venolia, G, Neiberg, F.: T-cube: a fast, self-disclosing pen-based alphabet. In: Proceedings of the SIGCHI conference on Human factors in computing systems - CHI 1994, pp. 265−270. ACM Press, Boston (1994)
23. Xiao, R., Laput, G., Harrison, C.: Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In: CHI 2014, pp. 193−196. ACM, Totonto (2014)