SVIFT: Swift Vision-Free Text-entry for Touch Screens

Aakar Gupta
University of Toronto
aakar@cs.toronto.edu

Navkar Samdaria
University of Toronto
navkar@cs.toronto.edu

ABSTRACT
We describe SVIFT – an eyes-free text-entry technique for mobile touch devices to increase speed and accuracy while enabling fast learning and easy adoption. The technique involves expanding the text-entry mode to a full screen format and innovating over the old-style T9 and telephone pad design. It makes use of multiple input and feedback interactions – swipe, pause, circle, hand-waving, audio and vibrotactile. A preliminary evaluation of two telephone pad based T9 prediction schemes found temporal speed and accuracy improvements in Svift.

DESIGN

Ingredients

SVIFT makes use of the existing hardware in the present touch phones and do not require any additional peripherals. Users use their fingers to interact with the touch screen and the accelerometer to give additional inputs and are given audio and vibrotactile feedback.

We make use of the concepts of existing techniques in our design, primarily Swype [1], T9 prediction [2] and positional vibrotactile feedback [3].

Target Users

Although the design is targeted for visually enabled users, especially in the case where they take a look at the text after completing typing a message; it has no design limitations for visually impaired users. However, considering that the visually impaired users are differently abled (eg. braille), SVIFT does not aim to compare against schemes designed specifically for them and limits its evaluative scope to visually enabled users.

Interface

This section discusses the interface design and the justifications behind the design decisions.

The design of the technique was grounded in the assumptions that existing input modalities are utilized for text-entry to enable fast learning. Plus, for complete utilization of the screen the typed text display box was done away with. Another concern was to enable the users to distinguish between buttons on the screen by knowing their finger's relative position on screen.

Keeping these considerations in mind, the screen is designed to contain a telephone keypad (Fig.1(a)). The telephone keypad enables large keys on the touch, accessible from boundary positions so as to allow the user to position on any key without looking with little practice. This allows for one-handed text entry. The screen remains the same in the landscape mode.

We make use of 5 types of interactions for input- a) Swipe b) Pause c) Circle d) Horizontal hand-wave e) Vertical hand-wave – the former 3 being touch and latter 2 being accelerometer interactions. Further we use two types of feedback mechanisms – a) Audio b) Vibrotactile.

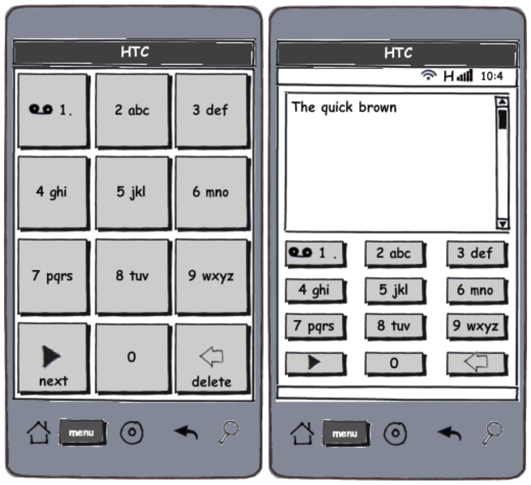

Fig1. Two screen view Mockups (a) Virtual Screen for Eyes-Free typing (b) Screen displayed when the user wants to view the message

Text-Entry

To write a word, the user can swipe his finger on the touch screen and pause at whatever character key he/she wants to type. Assuming that the user knows the position where he/she started writing on the screen (which is a valid assumption since the starting position can easily be related to the boundary position), the swiping gestures are guided by vibrotactile feedback whenever the user crosses a dead zone between two buttons. This is done to reinforce the user regarding the path they are following. The reason for not choosing audio feedback here was that there are other audio prompts (to be described later) which cater to characters being typed in. Vibrations were good ways of letting the users differentiate between swiping and other interactions.

Note that simply swiping over a key does not type a character. Swiping is required so that starting from a known position, the user knows where he/she is going. Employing lift and tap here might disorient the users of their position each time. A character is typed when user pauses at a button for a fraction of a second (the right duration will be determined after prototyping) – the typing in of a character is accompanied by an audio feedback which conveys the first letter of the letter group of the button. Note that this letter might be different from the letter that the user actually wants to type and only conveys the group identifier to let the user know that he/she is on the right key. Each pause is similar to the key press for a T9 input and as soon as the user lifts his/her finger from the screen after a sequence of swipes and pause, the word is assumed to be complete and the first word from the T9 dictionary (as there might be multiple words for the same combination of keys) is typed in. We did not select XT9 as word suggestion is not a part of our scheme.

The typed word is again spoken to the user using text-to-speech converters so that the user can decide to keep this word or choose other possible words which have been predicted. To go to another word from the predicted set, the user presses the *next* key at the bottom-left corner of the screen and goes on till he/she finds the required word.

In case when the combination of keys entered by the user does not return even a single word that exists in the dictionary, a vibrotactile feedback indicates that the input is incorrect (Though the word is still spoken to the user).

To delete a character, the user presses the *delete* key at the bottom-right corner of the screen; holding the key down deletes the entire word, both accompanied by different sounding audio feedbacks.

To register the same key twice, the user follows the technique employed by Swype which is to make a circular motion on the button after the first pause. The audio feedback again informs the user of his/her action.

To enter special characters, the user pauses at key *1* (top-mid key) and then goes through pressing *Next* to arrive at the character of choice, which is prompted to the user through audio.

Besides the default mode *Caps Off* (‘Abc’), there are two other typing modes – *Caps On* (‘ABC’) and *Num* (‘1-9’) which are changed by a horizontal wave of the hand and is recognized by the accelerometer and is prompted to the user through audio on change. The horizontal wave motion will be a complete cyclic motion; e.g. a wave in the right followed by one back in left. This is done so as to avoid mode changes by casual hand waves. Note that in the *Num* mode, the keys only type numeric values. This was done keeping in mind the problems users face while entering numbers in the telephone keypad as they have to pause at a key for a long time to get the numeric value; numbers are frequently entered in a message in a sequence to signify a phone number, for instance, and it make all the more sense to define a separate mode for numeric input.

For the case, when the user wants to see the message he/she has typed, a second view mode as shown in Fig.1(b) can be activated using a vertical hand wave detected by the accelerometer. An audio feedback is given when the switch occurs to inform the user so as to avoid any inadvertent switches.

User Scenario

Now, let us see a user scenario from the end-user’s perspective where the user ‘Bob’ wants to type in the string “Cat 2”. This scenario has been depicted in Fig2. Following are the set of actions and feedbacks that Bob goes through (The key will be referred by the numbers denoted by them. Eg. *2* for the top-mid button) –

1. Bob has not yet adapted to all the key positioning on screen and so he starts with a straight forward top-left corner.
2. Without pausing, he swipes his finger from *1* horizontally. He feels the vibrotactile feedback at the dead space between *1* and *2*.
3. He stops swiping when the vibration stops. He knows he is at *2*, which is required for the first character and therefore pauses there.
4. He receives audio prompt ‘*A*’, and understands that the first key is registered (‘C’ lies in ‘ABC’).
5. He now makes a circular motion on the key to register the second character from ‘ABC’ and receives another audio prompt ‘*A*’.
6. Next, he needs to go to *8* for ‘TUV’ and starts swiping vertically downwards. He feels the vibrotactile feedback at the dead space between *2* and *5*.
7. Without stopping, he crosses *5*, feels another vibrotactile feedback and pauses when the vibration stops.
8. He receives the audio prompt ‘*T*’ and lifts his finger up to indicate that the word is complete.
9. He receives audio prompt ‘*Act*’, and knows that he needs to go to another prediction.
10. He goes to the bottom-left corner of the screen and presses the *next* key.
11. He goes to the bottom-left corner of the screen and presses the *next* key.
12. He receives the audio prompt *‘Cat’* and knows that the word has been typed.
13. Next, he needs to type *2* (A space is automatically given after a word on the event when the next character is typed) and therefore does a complete horizontal wave motion.
14. He receives audio prompt ‘*Num*’ and knows that he is in the numeric mode.
15. By now, Bob feels confident of the positioning of *2* and guides his finger to where he perceives *2* might be and pauses there.
16. He receives the audio prompt ‘*3*’, from which he knows he was mistaken.

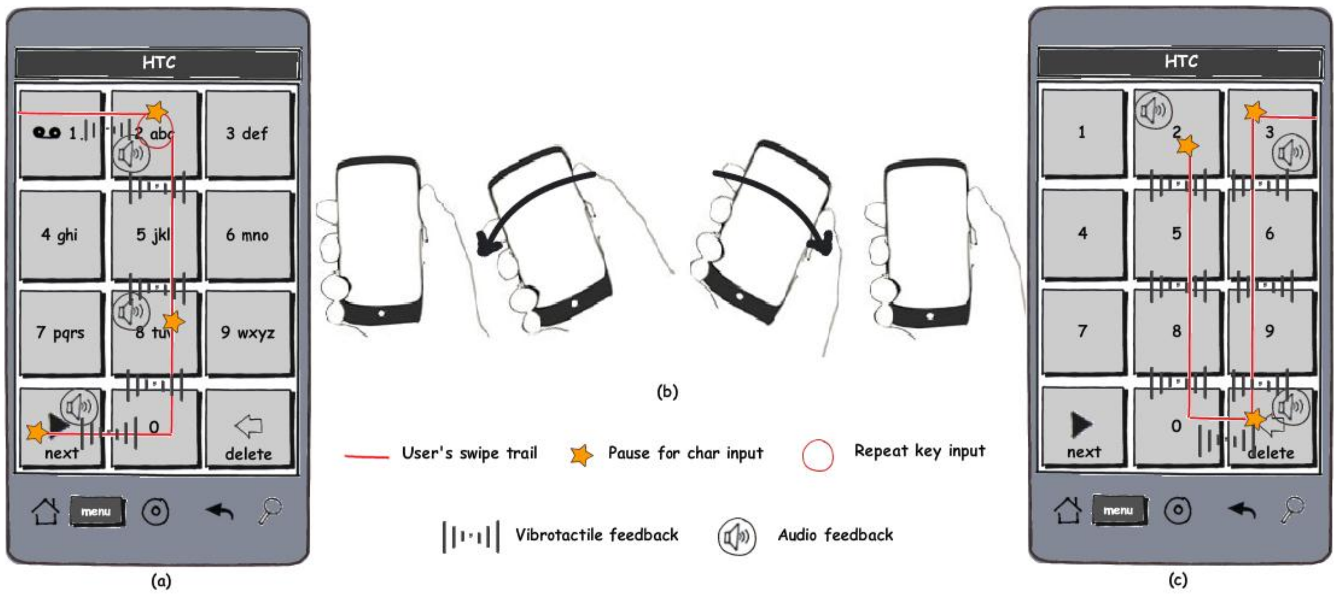

Fig. 2 User Scenario when a user types “Cat 2” in SVIFT (a) Types in *Cat,* by swiping and pausing (b)Horizontal Wave Motion for mode change to *Num* (c) Types in *2* after a mistake.

17. He then swipes downwards to the *delete* key with the aid of the vibrotactile feedback and pauses at the *delete* key. (Instead of swiping, he could have lifted his finger and pressed *delete* as well.)
18. He receives the audio tone for character deletion.
19. He then swipes leftwards and then upwards to arrive at *2* and pauses there.
20. Finally, the user uses the vertical wave to see text.

The text “Cat 2” is now written.

Strengths and Weaknesses

SVIFT utilizes a number of prior design concepts to its advantage in addressing eyes free text entry. Besides having a very fast learning curve (especially for users habituated in T9), the design promotes accuracy by giving feedbacks to the user at each and every step. Additionally it does not use any extra hardware and can be as easily adopted as downloading an app. It does not make use of any peripheral buttons in the phone, thus not interfering with other functionalities of the phone. Also, its single finger single hand usage increases the flexibility and ease of usage. For proficient users, the design encourages faster typing as there are no categories that the user has to go into and get out to type the A-Z letters unlike [4].

But, there are certain areas where the technique requires more refinements. As is the issue with the existing eyes-free techniques, the input mechanism for special characters is not efficient. Also, having numbers in a separate mode though could prove more effective in the long run; in the beginning it might lead to some confusion. For any T9 prediction method, the words that are initially not present in the dictionary pose teething troubles.

PROJECT IMPLEMENTATION

SVIFT is developed in Java over android 2.2 development platform and was evaluated on the HTC Desire smartphone. After completing the initial prototype we ran a small pilot study with 2-3 participants to get some general feedback on the features and usability of SVIFT. The results from this study were incorporated in our final prototype.

In the next phase we conducted an experiment to evaluate SVIFT against one of the existing eyes-free interaction techniques.

EXPERIMENT

We conducted an initial pilot study to understand any pertinent usability issues that the users face and made improvements in the prototype accordingly. Followed by this we conducted evaluative testing for our scheme and, to compare, we ran evaluations for an existing scheme as well. These are described in detail later.

Pilot Study

We conducted a pilot study with two participants where each of them were asked to type in four phrases each in eyes-on and eyes-free conditions. Eyes-on condition is defined as when the participant sees the phone screen constantly while typing. Eyes-free condition is when the participant is not allowed to see the phone screen while typing. Participants were asked to type in two random phrases and then give feedback on their experience. Besides routine bug-fixing, following were the primary outcomes of this phase – a) as we have mentioned, our scheme depends on button pauses for text entry. We had earlier kept this pause duration to be one second, to allow the user enough flexibility so as not to mistakenly pause on a button. Both the participants stated that they felt this duration was too long; as a result we changed the duration to 500ms which the users were more comfortable with. b) Secondly, the scheme was designed such that spaces between words are automatically entered when a participant lifts up his/her finger. This resulted in much confusion in combination with the T9 prediction and was consequently changed to a manual space. There were other insights which informed our experiment design and we will describe them in the context of their usage later.

Quantitative Experiment

We conducted a quantitative evaluative study of our scheme Svift and compared it with a modified version of an existing scheme proposed in [8]. This scheme is a simple T9 tap on a touch screen with nine keys which we have modified into a one with 12 keys and thus included the ability to go through multiple words for the same combination of letters and the ability to make error correction easy through the use of a delete key. Hereon, we will refer to this scheme as *T9 Tap*. The reason we selected T9 Tap as the one to compare against were multifold – 1) Svift was very similar to this scheme and the comparison would enable us to understand if there were any incremental performance benefits/hazards due to the variations which we had incorporated viz. swiping, pausing and vibrotactile feedback. 2) Due to limited time availability, it was easier to implement.

Participants

Considering time constraints, we decided to have four participants in a tightly controlled demographic of male users aged 20-25 who use touch inputs on a daily basis. All participants were right handed. We did not select a wide demographic of users, as given the less number of participants; we would not have been able to justify our inferences as they could result due to a variety of factors. As mentioned in the design document, the evaluation is conducted for visually enabled users. The participants were not paid; instead since the participants were lab students, they were happy to participate in return for the same favor from the authors for their respective research.

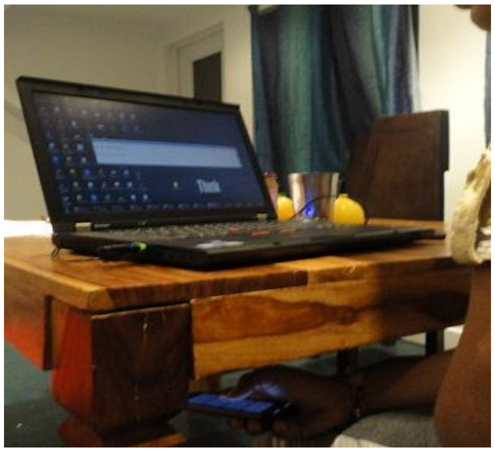

Fig1. Eyes-Free entry. Phone Screen is occluded.

Apparatus

A touch screen mobile phone was used for the application prototype while a computer screen in front displayed the English word phrases to be typed. Participants sat on a standard study chair with the computer screen at eye level. The interaction was one-handed and the participants were asked not to use the second hand for holding the phone. For eyes-free condition, participants were required to hold the device under the table, thus occluding it from view. A typical setup where the phone screen is hidden from the participant is shown in Fig 1.

Procedure

The experiment comprised of two schemes and hence for counterbalancing two participants were randomly selected for evaluation with Svift first and T9 Tap later (with a gap of at least one day between the two to counter fatigue effects) and vice versa for the remaining two. Next, we will describe the tasks that the participants performed.

For each scheme, the experiment proceeded in two parts – training and evaluation. The training phase involved an initial introduction to the scheme by the researchers to the participant (explanation of task and demonstration of software) followed by an eyes-on entry of four random phrases, further followed by eyes-free entry of four more random phrases. This was done as the pilot study had proved that eyes on followed by eyes-free training helps the participant immensely in building a mental visual model of the telephone keypad and developing a beginner' muscle memory for the same. A key takeaway from the pilot study which we made use of here was making use of the image of a telephone keypad on the computer screen in the eyes-free training mode for the first two phrases, so that the participant the transition from eyes-on to eyes-free becomes easy.

Training was followed by the evaluation phase where the participant did three contiguous blocks of eyes-free entry. Each block consisted of four phrases taken from the standard phrase-set [7]. Note that the phrases were same for all the participants, as well as the same for both the schemes. Participants were instructed to enter text "as quickly and accurately as possible".

Participants were encouraged to take a short break between phrases if they wished. The per-phrase data including time stamps, errors, deletions and re-entry were logged. Start timing for each phrase simultaneously began with the participant being informed to start typing and ended with the last word of the phrase being typed. The start time was not considered to be the first touch as this results in the loss of period that the participant has to spend finding the position of the first key.

Data Analysis Method

The total amount of entry was 4 participants x 2 schemes x 3 blocks x 4 phrases/block = 96 phrases. We acknowledge that this a small study; however our claims later are appropriately made, keeping this fact in mind. The primary parameters for performance evaluation were the standard - 1) Speed – words per minute (wpm) (1 word = no. of characters/5); 2) Accuracy – KSPC [6] (keystrokes per character). Each keystroke means a key pause in Svift and a tap in T9 Tap. The evaluation included the errors made by the participants such as a 'Delete' operation could mean that the character has to be retyped or an extra character has to be removed – both cases were accounted for. Also, if the word to be typed is the second T9 prediction instead of the first, the *next* key has to be pressed and this was also included as a key stroke.

Note that we have not used the MSD [9] (minimum string distance) method of evaluating the accuracy. This is because in the T9 mode, each key press can be construed as one of 3 or 4 characters which greatly increases the complexity of utilizing MSD in this scenario. To counter this, we instructed the participants to write the correct phrases each time and not leave a phrase if it is wrong, so that MSD is virtually zero (since MSD only reflects errors in the final text).

The variables to be tested for were a) the performance comparison of Svift v/s T9 Tap b) the performance curve as the blocks progressed

DATA ANALYSIS RESULTS AND DISCUSSION

Speed

The results for entry speed are shown in Fig 2. For Svift, entry speed increased significantly with practice (within the blocks) (Repeated-Measures ANOVA – F(1,3) = 9.276, $p<0.05$). The average entry speed for Svift after the third block is 4.75 wpm; while the overall mean is 4.18 wpm. Though the speed outcomes are low as compared with existing schemes in the literature [5], the steep learning curve of the scheme is impressive.

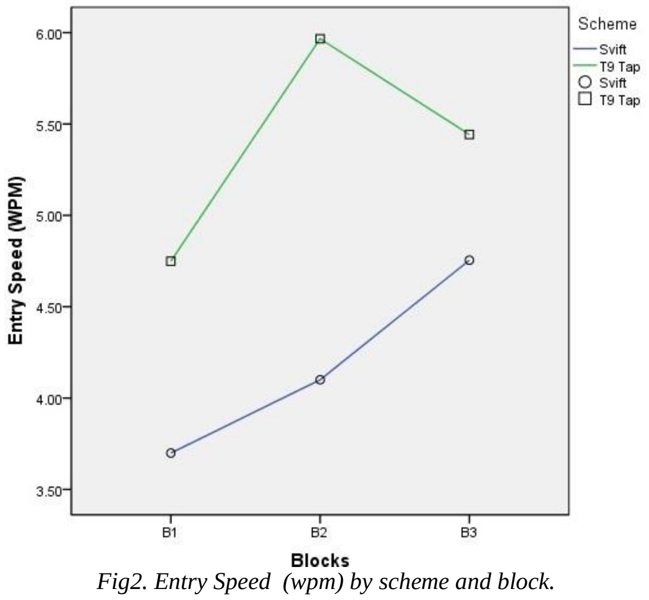

Fig2. Entry Speed (wpm) by scheme and block.

However, the same is not true for T9 Tap. No significant differences were observed in T9 Tap within the blocks. The average entry speed for T9 Tap at the end of third block was 5.44 wpm and the overall mean was 5.38 wpm. But, the entry speed difference between Svift and T9 Tap was found to be non-significant (although the figure shows difference between the entry speeds of the two schemes).

This leads to a few interesting points. There is no significant difference between the entry speeds of the two schemes at the end of the third block, which leads us to believe that in the initial stages, at least, the tactile modifications do not improve upon speed as desired. But given the steep learning in Svift, it calls for further investigation of the technique to look for speed improvements in future. One of the drawbacks of Svift, which can be construed as a reason for its similar speeds with T9 Tap, could be because of the pause duration limit of 500ms for each character. This limits the speed such that each character will at least take 500ms to be entered.

Accuracy

The KSPC analysis results are depicted in Fig3. The KSPC accuracy for Svift increased significantly with practice (within the blocks) (F(1,2) = 8.732, $p<0.05$). The average KSPC value at the end of three blocks is 1.07, while the overall mean KSPC was 1.17. This is a very remarkable outcome, given that the KSPC in existing schemes is close to 1.3. Of course, effects of a small tight participant set might have influenced these values, but it still gives another indication to further explore the scheme.

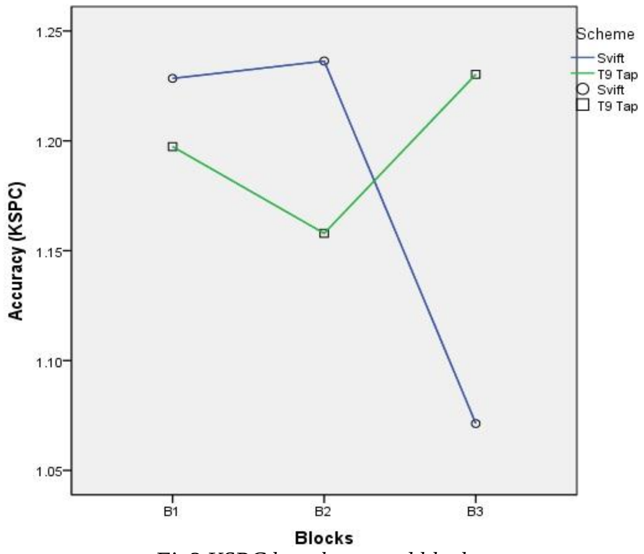

Fig3.KSPC by scheme and block.

Again, no significant effects were noticed in the T9 Tap within the blocks. The average KSPC value at the end of three blocks is 1.23, while the overall mean KSPC was 1.95. The difference in KSPC averages between the schemes is not significant.

The absence of learning in T9 Tap (for both speed and accuracy) indicates that there is no scope for improvement with this scheme, at least within the scope of limited trials that we have conducted. However, the learning rates were significant in Svift for both speed and accuracy (which is notable even at the third block) warrants further work.

After the testing, participant feedback on their experience was neutral with regards to both the schemes. All the participants felt that both the schemes were hard to use earlier but became easier as they proceeded. One of the participants mentioned the limitation of the 500ms pause as hindering his speed.

CONCLUSIONS AND FUTURE WORK

Following claims can be made - We evaluated two telephone pad based T9 prediction schemes and found temporal speed and accuracy improvements in Svift. We did not notice any significant difference in the average speeds and accuracy of both the schemes. This can be attributed to the fact that the two schemes are very close to each other structurally and a deep analysis is required to bring out the differences. The participant set was very limited and further work should address exhaustive evaluations. Also, text entry evaluation techniques beyond the ones employed here could shed more light on various relationships between the speed-accuracy trade-off.

There are some aspects of the prototype that we would want to change – the first one being removing the manual entry of the space key as it has been reported by participants that going to press the space key, after each word, distorts the mental model of the keypad. Secondly, we need to find a way to counter the pause duration which limits the speed. We noticed that the technique could very well be used by visually impaired users and would like to mold it to fit their requirements better.

REFERENCES

1. MacKenzie, I. S. KSPC (Keystrokes per Character) as a characteristic of text entry techniques, Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction - Mobile HCI 2002.
2. MacKenzie, I. S. and Soukoreff, R. W., Phrase sets for evaluating text entry techniques, CHI 2003.
3. Sánchez, J. and Aguayo, F., Mobile messenger for the blind, Proceedings of Universal Access in Ambient Intelligence Environments - 2006
4. Soukoreff, R. W. and MacKenzie, I. S., Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic, CHI 2001
5. Tinwala, H, & MacKenzie, I. S. (2009) Eyes-free text entry on a touchscreen phone. Proceedings of the IEEE Toronto International Conference – Science and Technology for Humanity – TIC-STH 2009, pp. 83-88. New York: IEEE. [C]
6. http://www.swypeinc.com/
7. http://www.t9.com/
8. Hall, M., Hoggan, E., and Brewster, S. T-Bars: towards tactile user interfaces for mobile touchscreens. MobileHCI, 2008.
9. Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, W. Keith Edwards. No-Look Notes: Accessible Eyes-Free Multi-touch Text Entry. Pervasive 2010