On The Efficiency Issues of Virtual Keyboard Design

Cihan Topal, Burak Benligiray
Department of Elec., Electronics Engineering
Anadolu University
Eskisehir, Turkey
{cihant, burakbenligiray}@anadolu.edu.tr

Cuneyt Akinlar
Department of Computer Engineering
Anadolu University
Eskisehir, Turkey
cakinlar@anadolu.edu.tr

***Abstract*—Virtual keyboards are useful tools, which ease effortless entry of textual data and enable typing with alternative input hardware such as a single switch. Early virtual keyboards are designed similar to physical keyboards in terms of appearance. Since a physical keyboard is designed for tactile-use; the usability of the first virtual keyboards comes up short in utilization with pointing devices. For this reason, more useful virtual keyboards are proposed with improvements in modal and functional properties. In this study, we examine a couple of design issues for virtual keyboards to provide their efficient utilization with pointing devices. We analyze the effects of visual key layout to the performance of virtual keyboards in connection with the statistical properties of the target language's vocabulary. We also propose a virtual keyboard design for a comfortable text entry experience based on our observations.**

Keywords- virtual keyboard, text entry, user interfaces, assistive technologies, letter prediction.

I. INTRODUCTION

Human-computer interface is a field that expands upon both software and hardware design. Currently, the most common way of human-computer interaction consists of using a keyboard and a mouse to input data, and looking at a screen to interpret the output data. Whenever a computer has a human client, such input-output hardware becomes a necessity. Nonetheless, a screen, a mouse and a keyboard is certainly not practical for all kinds of applications. Either the computer may be so small that the input-output hardware overhead becomes too costly, or the user may not be able to operate such hardware because of their physical condition. For some similar cases, importing features of the human-computer interface from hardware level to software level will be sufficient. One of the most common examples of this application are on-screen keyboards that removes the need for a physical keyboard.

On-screen keyboards (OSK) are being used for many purposes; to improve accessibility, reinforce security while entering critical data, or as an alternative to physical keyboards where they are not viable anymore, such as the case for smartphones. Consequently, there are many distinctions regarding the design approach, as well as the performance metrics. While our study predominantly focuses on accessibility, we derive from various fields of keyboard design studies, and believe that our proposals will be beneficial to all OSK designers looking for a different approach in OSK design.

II. RELATED WORK

The most decisive, if not limiting factor in OSK design, is the intended input hardware. The improving cost-efficiency of sensor and computation technologies allows us to design OSK for cutting-edge input hardware. However, there always will be demand for simpler input hardware, as severity of users' disabilities may prevent them from using more demanding hardware. The simplest form of input to an OSK is a single switch. There may be variations in the type and number of this switch: [1] uses two touch switches instead of one, while [2] recognizes the user's winks as the action on the switch. The majority of such hardware is used with scanning keyboards, although alternatives such as using Morse code are possible. [1] also illustrates that displaying large numbers of characters on screen is feasible when discrete input is being used. If the input was continuous, displaying many characters would require unrealistic amounts of precision from both the hardware and the user.

Just as a particular input hardware may be used for different kinds of OSK, there are some OSKs that may be used by different kinds of input hardware, as in [3, 4]. However, the design considerations for various input devices are rather different. For example, button shapes will be of little relevance if the OSK will be used with a discrete input device, while they will be critical when the OSK is used with a continuous input device. Another difference between the input types is the effect of using hierarchy. Assuming no prediction or completion, when no hierarchy is applied, the user will do one selection per character entered. When hierarchy is applied, characters are grouped together, and the user chooses the group and the character, resulting in more than one selection per character. [5] shows that using hierarchy for discrete input keyboards is beneficial. Continuous input devices have a map of inputs that consists of virtually infinite possible elements per action. On the other hand, the map of discrete inputs has the same number of elements as the possible actions (i.e. the number of switches). While expanding the keyboard layout to sides slightly enlarges this map in one dimension, adding a hierarchical group adds a new dimension to the input map. Accordingly, a synthesis of hierarchical grouping and choosing the optimum number of characters for each group will yield the best possible solution.

We mention that multi-modal keyboards may not be designed to use their input hardware potential to the fullest. We also discussed that while applying hierarchy is beneficial for OSK to be used with discrete input devices, it is not needed for

This work is partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with the grant number 111E053.

978-1-4577-1759-8/12/$26.00 ©2012 IEEE

OSK that is operated with continuous input, as the possible inputs per each selection is virtually infinite. An example of this situation is clearly demonstrated in [3, 6-7]. While gaze tracking hardware becomes more precise as time progresses, these studies propose hierarchical keyboards to be used with 2D continuous input. Instead of investing on the relatively high resolution of continuous input, hierarchical layouts are proposed, which are actually more beneficial for lower resolution (discrete) inputs. However, as [8] supports, it will be beneficial for the continuous input OSK to have at least one level of hierarchy. The keyboard chooses the common hierarchical group by default for each character, and the user switches the group if they want to enter a character from the other group. More frequently used characters will be in the default group, and the less frequently used characters will be on the second group. The off-screen targets proposed in [9] may be used for OSK designed for gaze interface to switch between hierarchical groups.

Scanning keyboards impose a minimum time to enter a character, as the user has to wait for the scanning indicator to highlight the chosen character. On the other hand, selection options for continuous input are various. [10] lists the selection options for gaze interface as winking (or another predetermined mimic), hovering over the section for a predetermined time, or using an extra discrete input (like a switch). Furthermore, a gaze gesture is defined for selecting the character to be entered. Such gaze gesture-based systems are also proposed in other studies, as in [11]. However, none of these systems seem to be used by any practical applications, probably due to the users' reluctance to learn a completely new input system. Conventional keyboard types and layouts seem to be more preferable to the users. Even intuitive layouts such as alphabetical listing cannot perform comparable with QWERTY layout. More interestingly, a box shaped QWERTY keyboard where buttons are more reachable still cannot compete with the regular QWERTY keyboard [12]. This shows that not breaking users' habits unnecessarily is an important design issue. This also becomes apparent in [13], where test subjects complain about how characters do not have constant places on the layout. There are studies that explicitly state the importance of this problem [6], and having a constant keyboard layout is standard for most OSK.

Layout performance measurement is studied by using mathematical modeling of errors, counting expected amounts of activity per character, predicting average Words Per Minute (WPM) or verbally stating common design rules [3, 14-17]. The mathematical models are especially not portable to be used for different keyboard types, as they stem from studies done for physical keyboards. Many OSK use an alphabetical layout [4, 6] while there are others that use layouts optimized according to the character frequency in a corpus [3]. Since disabled users are not familiar with QWERTY layout of physical keyboards, it is no longer an unbeatable competitor, so alphabetical layout takes the dominance. However, alphabetical keyboards limit expert users by not being optimized. Although novice users will type better with alphabetical layout, expert users are expected to know the location of every character on the layout, so the alphabetical layout will be a crutch that will hinder expert users' typing speed.

Unlike physical keyboards, OSK can have many shapes, and may react dynamically to the user. There are OSKs that keep the traditional matrix shape [1, 7], and there are also circle shaped OSKs [3, 6]. While the circle shape offers better usability, the matrix shape uses the screen more efficiently. It could be argued that when the OSK covers the entire screen (e.g. screen of a GPS device), matrix shape is preferable. Most OSKs use word prediction to decrease the number of actions needed per character. The words generated by the prediction engine are offered to the user in a predetermined area [7, 18]. This actually causes the user to look at the prediction area and go back if the prediction is not true, which are separate actions if the user is using a gaze interface. The dynamic potential of the OSK can be used to give feedback to the user as well. [19] claims that the users type fastest with both auditory and visual feedback (compared to their absence). The characters that are expected to be entered next are highlighted in [18].

III. EFFICIENCY ANALYSIS OF VIRTUAL KEYBOARDS

In this section we talk about some efficiency issues we observed during experimentation with some existing virtual keyboards. To make a clear analysis, we can split the "typing a key" event into three logical stages, i.e., locating, dragging and stroking (or dwelling for continuous-input OSKs). Hence the total time for a typing event becomes $t_{locating} + t_{dragging} + t_{stroking}$. The first stage, locating, requires mental attention and takes some time to find the key to be pressed on the keyboard unless the user is experienced. When the key is located, the user drags the cursor or moves a finger on top of it; finally, the key is pressed.

A. Modal Issues on Efficiency

People can input very fast with a physical keyboard because they use multiple fingers in a serial way. Since the locations of the keys are memorized with experience, $t_{locating}$ decreases and the input speed significantly increases. While a finger is typing a letter, another finger moves on to the next key and can press the key very quickly right after the previous keystroke. Thus, stroking and locating stages of two consecutive typing events are overlapped and thus, the typing is partially pipelined as illustrated in Fig. 1. Obviously, $t_{stroking}$ is shorter for physical keyboards since there is no need to dwell on the key in contrast to continuous input OSKs.

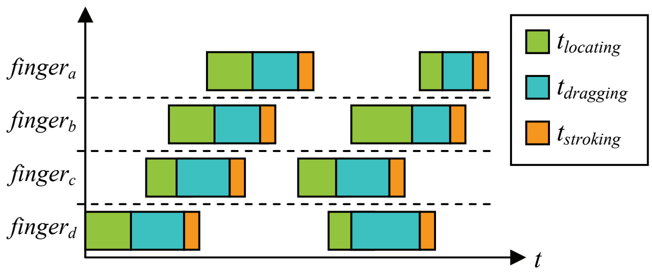

Figure 1. Timing illustration of typing events for multi-finger input.

Contrary to physical keyboards, partially parallel typing is not possible for OSKs due to the unity of the input medium. In other words, because a pointing device can point to only one location, the user has to drag the cursor after all of the three stages of a typing event are completed (see Figure 2). For this reason, key typing stages cannot be partially pipelined and time losses occur between consecutive keystrokes. These time losses can be minimized as the user gets experienced; however, there is still a time loss for dragging the pointing device from the current key to next one. If the user is novice or the key locations are dynamic, time spent for locating the next key takes longer.

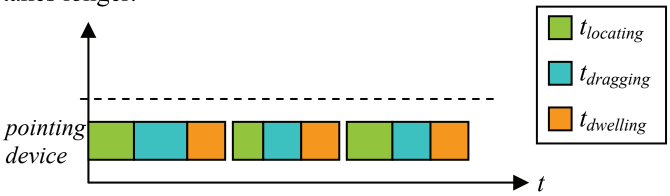

Figure 2. Timing illustration of typing events for a pointing device.

There are several ways to shorten the time between two typing events. The most useful method for this purpose is to use fixed locations for the letters. Thus, users can memorize the key locations by time and will not need to spend time for seeking the next key. By fixing the key locations on the keyboard layout, $t_{locating}$ can be decreased significantly.

Besides fixing the letter locations, there are also other improvements related to the locations of letters. For instance, to shorten the dragging time of the cursor from one key to the next one ($t_{dragging}$), the main layout of the keyboard should be organized in a way to minimize the average distance between the letters. This can be achieved by employing a compact, that is to say, a circular layout. OSKs that have rectangular layouts similar to physical keyboards complicate locating the keys with pointing devices. As a further improvement, letters that are more frequent in the target language should be placed closer to each other on the keyboard to shorten $t_{dragging}$. In other words, different priorities can be given to the letters with respect to their occurrence frequencies to increase the text entry throughput.

B. *Functional Issues on Efficiency*

Besides the model characteristics of a virtual keyboard, functional properties can also affect the text throughput. Common functional properties that are employed in virtual keyboards are automatic word completion and word prediction mechanisms. By the integration of a dictionary or a corpus with a proper algorithm, virtual keyboards can be made to predict the rest of the word. Thus, text entry process gets accelerated by typing prefix of a word rather than typing all the letters. Furthermore, in sophisticated systems, employed dictionaries can be adaptive and learn new words during the use. Therefore, more successful predictions can be made in order to provide more text input with less effort.

There are several methods to generate word predictions during the text input. The number of predictions can be very large, and tens of words can be suggested with respect to the already entered prefix. However, the limiting factor in providing word suggestions is the screen area in an OSK. In other words, increasing the number of word predictions results in decreasing the effective area of each word cell. This situation hinders the usability of the OSK when the accuracy of the employed pointing device is limited. For instance, with a low resolution eye tracker, the text throughput prominently decreases due to the involuntary gaze fixations on false cells. For this reason, in GazeTalk [7], all word predictions are listed in one large cell. If the user wants to choose one of the suggested words, they first select the cell that contains predictions to expand it. Then a second selection is required to select one of the suggested words. As a result, the utilization of the screen area in an efficient way becomes a crucial design issue of a virtual keyboard.

Eventually, we can summarize both the modal and functional issues that improve the efficiency of a virtual keyboard as the following;

- Locations of the letters should be fixed,
- Average distance between each key couple should be minimized (compact layout),
- Letters that have high occurrence frequency should be closer to each other,
- Available screen area should be used in an efficient way (splitting the area according to the probability of occurrence of the letters).

IV. A Virtual Keyboard Design Example

In this section we propose a virtual keyboard layout according to the aforementioned design issues in the previous section. Our keyboard does not need an additional event like a click and works with hovering. To keep the general layout of the keyboard compact, we preferred a circular structure for the placement of the letters (see Fig. 3). We divide the circle into three main parts; inner circle, middle ring and outer ring.

Middle and outer rings are reserved for the letter characters. Note that the width of the middle ring is greater than the outer ring for the compensation of cell areas. We call the regions remaining in the corners as complementary regions since they complement the main keyboard layout to a square. Complementary regions are employed for additional tasks that have secondary importance for text entry. The inner circle is used to display the current word prefix. Since it is at the center of the letters, the user can check the current state of the entered text very easily.

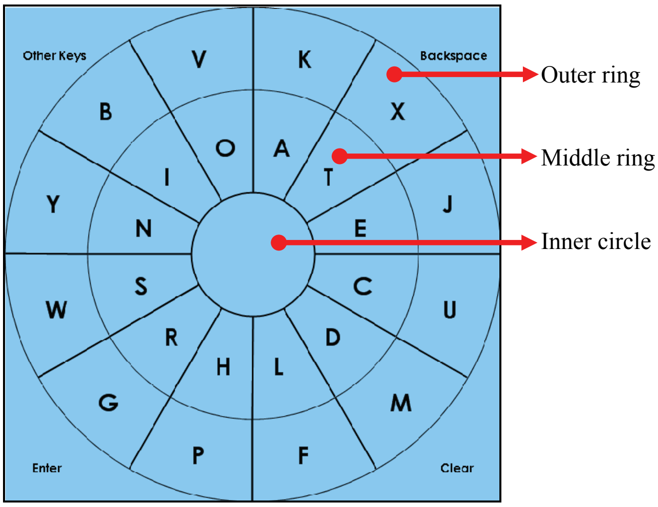

Figure 3. Layout of the proposed virtual keyboard design.

With this compact layout, we are able to keep all the letters close to each other in contrast to rectangular layouts. Moreover, we place the letters with respect to their occurrence frequencies to further improve the efficiency. We used a corpus consisting of English words with 450 million letters in total. We examine the distribution of this corpus and placed half of the most frequent letters into the middle ring in an attempt to keep them even closer. In Fig. 4 we present the distributions of letters in the corpus. We place the all letters in English alphabet except two least frequent ones, which are 'Q' and 'Z'. We prefer to place them under the "Other Keys" page of the keyboard rather than decrease the area of each cell. We also tried to standardize the total probability of each slice on the keyboard by matching the most frequent letters with the least frequent ones. For instance, we place 'E' and 'J' or 'T' and 'X' in the same slice. Therefore, we optimize the modal issues to increase the efficient usability of the keyboard.

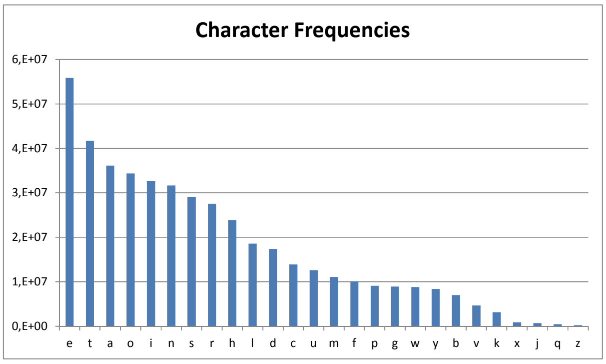

Figure 4. Character frequencies of English letters in the employed corpus.

Placement of the letters is not the only design property that utilizes the corpus. During the text entry, the keyboard uses the corpus to dim the letters that can not be the next letter according to the already entered prefix. Thus the perception of the user can be focused on the available letters in an easy manner. If the word that the user wants to enter does not exist in the corpus; the user is still able to select the dimmed letters. This situation is clearly shown in the Fig. 5.

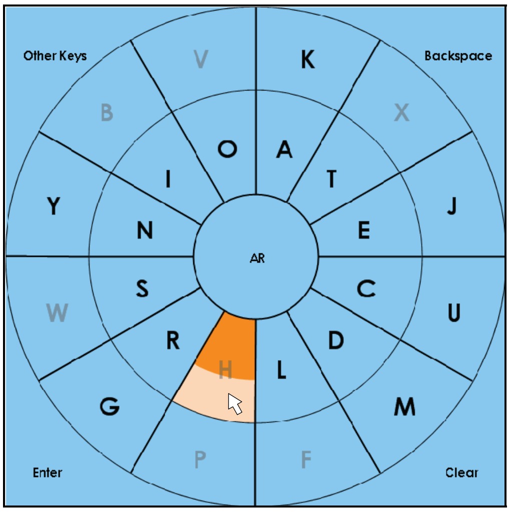

Figure 5. The potential letters for the prefix "AR" displayed in solid black and the others are dimmed. Note that the dimmed letters can still be selected ('H' in the example).

Another useful property of the proposed virtual keyboard is the word prediction suggestion strategy. As we mentioned before, the proposed keyboard works by dwelling without the need for an extra event. When the user moves the cursor of the pointing device, the cell starts to fill with an adjustable speed (see the cell of 'H' in Fig. 5). Each time the user selects a valid (not dimmed) letter, the keyboard suggests a word that starts with the current prefix in the cell of the current letter. In Fig. 6, the already entered prefix is "COM" as indicated in the inner circle and the cursor is hovering on the letter 'P'. As seen in the cell of letter 'P', the keyboard suggests the word "COMPANY" as the prediction that starts with "COM" and continues with 'P'.

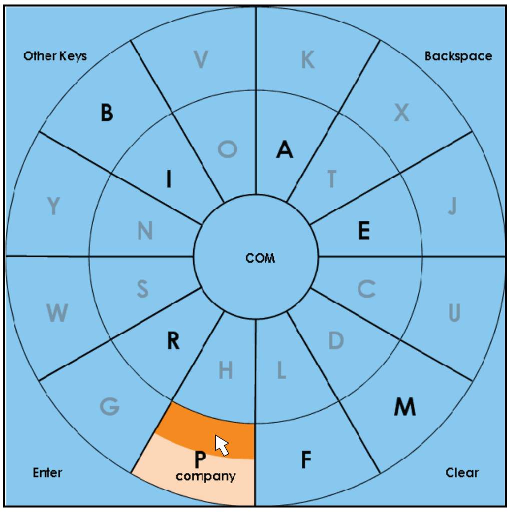

Figure 6. Keyboard suggests the "COMPANY" word for the prefix "COM".

If the user wants to use this prediction, the only action they are supposed to perform is to keep waiting on that cell. When the operation of filling with orange color is completed, another filling operation with green color will be automatically launched as seen in Fig. 7. If the user does not move out the cursor from that cell, the word "COMPANY" will be inserted to the text stream of the system.

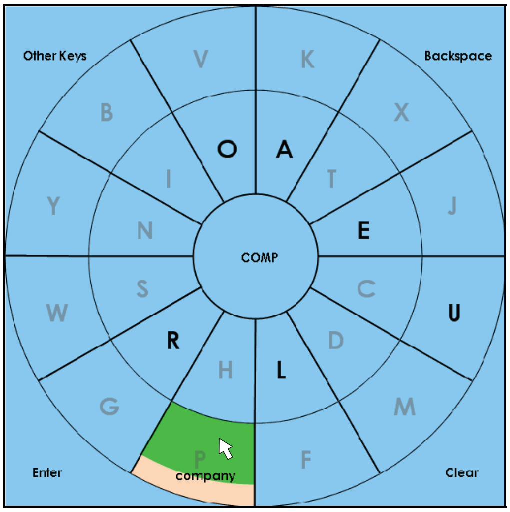

Figure 7. Automatic word completion mechanism of the proposed virtual keyboard.

Thus, word predictions can be suggested to the user in a very efficient way; and the user can select the suggested words effortlessly. As another advantage of this layout is the ability to suggest many word predictions, i.e., one prediction for each active cell, becomes possible. In other words, if the user moves the cursor to other not-dimmed letters, they will see another word suggestion starting with the current prefix in each cell. Our virtual keyboard is available at [20].

V. EXPERIMENTAL RESULTS

The experimental results are hard to replicate due to the human factor. Instead of testing only with the proposed keyboard, we also performed tests with two widely-used OSK, i.e., Dasher [4] and GazeTalk [7]. Keyboard options and parameters were not changed for any of the OSK. To be fair, our keyboard's dwell time was adjusted as the same as GazeTalk's default, which is 1000 milliseconds. The proposed keyboard is intended to be used in English, so the experiment was conducted in English. Twelve non-native speakers were employed as the test subjects. All of the subjects were novice for all employed OSKs and they typed a sample text with a mouse using each OSK for two minutes. To avoid the bias of getting experienced towards end of the experiments, the usage order of the keyboards was permutated. The average of individual results is presented in Table 1 in terms of words per minutes.

TABLE I. RESULTS OF EXPERIMENTS PERFORMED WITH OSKS.

Employed OSKs	Dasher	GazeTalk	Proposed
Word Throughput (Word per Minute)	3,50	3,07	5,42

By examining Table 1, we can observe that the word entry speed is highest for the proposed keyboard. Since the users were novice, the results indicate how intuitive the keyboards are. Most users felt intimidated by Dasher's unique design; while some users were able to pick it up quickly, others failed to grasp the fundamental concepts of writing with Dasher. On the other hand, users were quick to adapt to GazeTalk due to the fact that it has only ten selection options on the screen. However, this same property limits the input speed. Typing with the proposed OSK was generally easy to learn like GazeTalk, but the larger amount of selection options and the ease of using word completion caused the users to type relatively faster. Interested readers are free to download and use the proposed OSK from our website [20].

VI. CONCLUSIONS

In this study, we argue the design issues for efficient virtual keyboard systems and propose a design under the light of aforementioned properties. We implement our keyboard and test its efficiency with experiments. In the tests performed with novice users, we observe that the users can easily learn and use the system very quickly. The major improvements of our virtual keyboard can be summarized as, compact layout, effective placement of letters, ease of use and efficient word completion system. Interested users are free to download and use our keyboard with the choice of using their own corpus to train.

REFERENCES

[1] A. Mukherjee, S. Bhattacharya, P. Haider, and A. Basu, "A virtual predictive keyboard as a learning aid for people with neuro-motor disorders," Advanced Learning Technologies (ICALT) Fifth IEEE International Conference on , vol., no., pp. 1032- 1036, 5-8 July 2005.

[2] B. Ashtiani and I.S. MacKenzie, "BlinkWrite2: an improved text entry method using eye blinks," in Proc. ETRA, 2010, pp.339-345.

[3] V. Prabhu, and G. Prasad, "Designing a virtual keyboard with multi-modal access for people with disabilities," Information and Communication Technologies (WICT), 2011 World Congress on , vol., no., pp.1133-1138, 11-14 Dec. 2011.

[4] S.A. Wills and D.J.C. Mackay, "DASHER-an efficient writing system for brain-computer interfaces?", IEEE Trans. Neural Syst. Rehabil. Eng, vol. 14, no. 2, pp.244 - 246 , 2006.

[5] D. Kar, A. Pal, C. Bhaumik, J. Shukla, and S. Ghoshdastidar, "A Novel On-Screen Keyboard for Hierarchical Navigation with Reduced Number of Key Strokes," in Proc. SMC, 2009, pp.5072-5076.

[6] A. Huckauf and M.H. Urbina, "Gazing with pEYEs: towards a universal input for various applications," in Proc. ETRA, 2008, pp.51-54.

[7] J.P. Hansen, A.S. Johansen, D.W. Hansen, K. Ito, and S. Mashino, "Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections," in Proc. INTERACT, 2003.

[8] Jinan Guan; Yaguang Chen; Jiarui Lin; , "Designing a dual page virtual keyboard for mental speller," Proc. Int'l Conf. Neural Interface and Control, vol., no., pp. 55- 59, 26-28, May 2005.

[9] P. Isokoski, "Text input methods for eye trackers using off-screen targets", in Proc. ETRA, 2000, pp.15-21.

[10] T. Ohno, "Features of Eye Gaze Interface for Selection Tasks", in Proc. APCHI, 1998, pp.176-182.

[11] M. Porta and M. Turina, "Eye-S: a full-screen input modality for pure eye-based communication", in Proc. ETRA, 2008, pp.27-34.

[12] Coleman, M.F.; Loring, B.A.; Wiklund, M.E.; , "User performance on typing tasks involving reduced-size, touch screen keyboards," Vehicle Navigation and Information Sys. Conf., vol.2, pp.543-549, Oct. 1991.

[13] J.P. Hansen, A.S. Johansen, D.W. Hansen, K. Itoh, and S. Mashino, " Language technology in a predictive, restricted on-screen keyboard with ambiguous layout for severely disabled people", iIn Proceedings of EACL Workshop on Language Modeling for Text Entry Methods, 2003.

[14] Bhattacharya, S.; Basu, A.; Samanta, D.; , "Computational Modeling of User Errors for the Design of Virtual Scanning Keyboards," IEEE Trans. Neural Sys. and Rehabilitation Eng., vol.16(4), pp.400-409, Aug. 2008.

[15] Jain, S.; Bhattacharya, S.; , "Virtual keyboard layout optimization," Students' Technology Symposium, IEEE, vol., pp.312-317, April 2010.

[16] Sarcar, S.; Ghosh, S.; Saha, P.K.; Samanta, D.; , "Virtual keyboard design: State of the arts and research issues," Students' Technology Symposium (TechSym), IEEE , vol., no., pp.289-299, 3-4 April 2010.

[17] Kumar Sharma, M.; Dey, S.; Kumar Saha, P.; Samanta, D.; , "Parameters effecting the predictive virtual keyboard," Students' Technology Symposium (TechSym), IEEE , vol., no., pp.268-275, 3-4 April 2010.

[18] I.S. MacKenzie and X. Zhang, "Eye typing using word and letter prediction and a fixation algorithm", in Proc. ETRA, 2008, pp.55-58.

[19] P. Majaranta, I.S. MacKenzie, A. Aula, and K. Räihä, "Auditory and visual feedback during eye typing," in Proc. CHI Extended Abstracts, 2003, pp.766-767.

[20] AU CVLab Website: http://ceng.anadolu.edu.tr/cv/osk