Design, implementation and evaluation of a dynamic layout of a vision based virtual keyboard

Asif Azhar

University of Tampere

School of Information Sciences

Computer Science / Interactive Technology

Asif Azhar: Design, implementation and evaluation of a dynamic layout of a vision based virtual keyboard

M.Sc. thesis, 39 pages, 3 index and 7 appendix pages

May 2016

A hands-free text entry system is needed when the typical text entry with hands is not feasible due to the user's physical disability or other limitations. Use of head/face tracking is one of the options to interact with virtual keyboards for hands-free text entry. Performance and usability impacts of the layout of the virtual keyboard used with such hands-free text entry systems have not been studied enough. This thesis introduced a novice layout design of virtual keyboard to be used with face/head input. The aim of this thesis was to check if the performance of the new and traditional layouts will be any different. The new layout was inspired by Fitts' law. In the new layout, the size of each key was calculated dynamically in proportion to its distance from the last pressed key. The performance of the new layout was tested against the traditional static QWERTY layout in a user experiment with 16 able bodied participants where each user entered 8 text phrases of approximately 30 characters with each layout. Face tracking was used to control the cursor movement and a key from the physical keyboard was used to enter the selected character. Text entry speed was 5.03 and 5.14 words per minute, error rate was 0.83% and 1.28% for dynamic and static layouts, correspondently. Keystrokes per character was 1.05 with both layouts. Statistical analysis did not show significant differences in the performance of these two layouts. The subjective rating revealed that the participants liked both layouts equally but felt that the dynamic layout requires more mental effort and is less accurate than the static layout. Directions for further improvement of the dynamic layout are suggested as a future work.

Key words and terms: Computer vision, video based interaction, hands-free text entry, dynamic layout, virtual keyboard.

Contents

Appendices

1. Introduction

Touch operated on-screen keyboards, also known as *virtual keyboards*, have become more widespread as touch enabled devices including smart phones, tablets and others have gained in popularity. The task of *text entry using a virtual keyboard* has two integral parts: accessing the target key on the virtual keyboard and issuing a single "press" command to enter a desired character. Virtual keyboards have a number of important advantages. The use of a virtual keyboard instead of a physical one allows to have a large display size while keeping the overall device size relatively small. Although virtual keyboards in comparison to their physical counterparts are considered as less efficient [1], it is possible to re-arrange the layout, key size and shape of a keyboard layout at a run time to make the virtual keyboard better suited for a given situation. Therefore, improving the performance of virtual keyboards in terms of typing speed and accuracy has been a major area of interest in the field of human computer interaction (HCI).

Text entry performance with virtual keyboards further decreases when operating such devices using some other input method than traditional touch or mouse click [1]. This can be the case when the user has limited or restricted access to the device, for example, in case of physical disability of the user or when the device is used at places where physical interaction with the device is not possible or is not safe. In such cases conventional text entry methods of using hands to touch or hit the keys of the keyboard to enter characters are not feasible. Therefore, other options have been considered for their applicability to control virtual keyboards in the settings of hands-free HCI.

Physically impaired users may not have limbs or have a reduced accuracy of limb motor control and small muscle power to operate efficiently with physical keyboards, switch triggers and touch screens. However, many of these users preserve control over their gaze, face and neck [2]. For such users, eye tracking and head/face analysis based interaction are the alternative options to enter electronic text without the use of hands. In so-called *eye typing*, the user is looking at the keys of a virtual keyboard and dwells his/her gaze on a key for 500-1000 milliseconds (ms) in order to "press" it. Eye typing has been quite intensively investigated in the past [2]. Arguably, one of the largest constrains of eye typing is that eyes are fully occupied by the typing task itself, although

the eyes biologically have been developed for free observation and collecting visual information from the environment and not as a control mechanism [3]. For this reason, users may experience difficulties while entering text by gaze and simultaneously performing other tasks that require visual attention such as, for example, interaction with other applications on the computer desktop or objects in the physical environment.

The idea of *face typing* is to monitor the movements of a selected feature of the user (head, face or individual facial feature), extract recurring patterns from the monitored movements and generate control commands accordingly [4] [5]. In hands-free text entry, the tracked movement of head or face can be used to control an on-screen pointer (usually called a cursor) to access a desired key on the virtual keyboard. By definition, head tracking based methods of pointing at the elements of the virtual keyboard overcome the usability limitation of eye tracking technology in control demanding tasks as they allow users to freely observe the surroundings in performing various tasks. Moreover, the analysis of information of the user's face and head offers more functionality to computer control. Altogether there are 43 muscles in the human face allowing to generate a variability of distinct expressions. Thus, facial expressions and head gestures can be potentially used for execution of and switching between different interface options, for example, to issue single or double click on the virtual keyboard. Noteworthy, these operations do not require expensive hardware equipment. Head or face movements can be relatively easily tracked with face tracking software (SW) using stream of camera images; and this technology is constantly improving in its accuracy and robustness to environmental factors. Therefore, text entry that is based on the analysis of head movements and facial expressions of the user is a feasible and affordable solution of hands-free text typing in the situations when conventional text typing is not possible.

This thesis work focuses on a new and relatively unexplored technology of head/face based control of a virtual keyboard for hands-free text entry. The motivation of the thesis comes from the fact that earlier studies related to face typing had focused more on technical aspects of different methods used for automatic face analysis [5]. Researchers have been analyzing different head/face tracking technologies as well as the applicability of using different face expressions to control virtual keyboards. There has not been much

work done to study the impact of the new input modality on effectiveness, efficiency and user's satisfaction of text entry.

Obviously, text entry has an increased difficulty and latency in hands-free operation as compared to the case of using hands to perform text entry. Therefore, next important consideration is that the layout of a virtual keyboard is an important factor in text typing [6]. As it will be explained in the next section, most of the past work has been done using static, QUERTY or alphabetical, keyboard layouts [5] [4]. It appears that only few optimizations have been done in designing virtual keyboards for face/head input [7] [8]. However, these solutions may not be considered optimal as they often require two "clicks", or *keystrokes*, in order to type a single character. Other layouts optimized for touch or gaze based text typing may not be suitable for face typing [9] [6]. Therefore, it is important to investigate a better suited keyboard layout for face/head input as a pointing mechanism in text entry tasks.

The main contribution of this thesis work involved designing, implementing and testing a new layout of an on-screen virtual keyboard. The design of the proposed virtual keyboard layout was inspired by Fitts law [10] [11][ISO 9241-9:2000]. Fitts law is well established for its applicability to user interface (UI) designs. It predicts that the time required to quickly reach to a target element is a component of the proportion between the distance to the element and the width of the element [10]. Unlike typing with hands or gaze, in face typing it takes longer time to move cursor from one key to the next target key [5]. Based on Fitts law, increasing the size (width) of the distant keys is hypothesized to ease the access of such distant keys. The new dynamic layout proposed in this thesis work does this. The keyboard layout dynamically adjusts itself as user enters text. With each pressed key, the size of keys changes as a factor of distance from the pressed key making the distant keys larger and therefore easier to access. Performance of this newly designed dynamic keyboard layout was tested against the traditional QWERTY keyboard in a user study using face tracking for text typing.

This thesis is divided into six chapters. Chapter 2 contains a literature review of the previous work related to hands-free text entry. Chapter 3 covers the design and implementation details of the developed keyboard layout. Chapter 4 describes the details of the user study performed to compare the performance of the new layout against the

traditional QWERTY layout. Chapter 5 lists the results of the experiment. Chapter 6 discusses the results. Final conclusion and future options are presented in chapter 7.

2. Literature Review

Several studies have investigated different aspects of using virtual keyboards for hands-free text entry. These include both using gaze and face/head movement tracking for cursor control. This chapter outlines some of these prior works that include a proper user study with empirical evaluation of user typing performance and satisfaction. It should be noted, however, that the results of individual studies reported in the literature cannot be directly compared due to a number of factors like different experimental setups, different layouts of virtual keyboards, etc. For example, it is not so straightforward to evaluate the performance of keyboards with predictive text. The result depends on the hit-rate of prediction. Performance would be poorer if the words-to-write are not found in the used language model of the system. This can be the case, for example, when writing involves lot of names, words from other languages or abbreviations. This limitation is in general valid for all prediction based writing systems. Another aspect is that any layout other than the well-familiar QWERTY layout does have a learning curve for novice users. Despite of these limitations, the review provides important insights into the design of virtual keyboards optimized specifically for face/head input.

Špakov and Majaranta [9] pursued the objective of reducing screen space covered by on-screen virtual keyboards in their study. They used eye tracking as input method in this work. Their work involved a novice and interesting keyboard layout. They first discussed the layout ideas presented in previous studies on a similar topic [12] [13] and noted that all these systems conserve screen area but learning the *gesture-based input* (where gaze trail is followed and a certain trail pattern is associated with a certain character) takes time. Furthermore, these systems require several (typically 2–4) *keystrokes* per character (amount of key-presses needed to enter single character). They cited an average typing speed with these systems to be 5–8 *words per minute* (wpm, 1 word has 5 characters including the space and punctuation marks) [14] using gaze control. In their study, Špakov and Majaranta proposed a scrollable virtual keyboard to be used for hands free text entry. The motivation behind the keyboard layout design was to reduce the on-screen space consumed by the virtual keyboard and yet to keep the familiar QWERTY layout without reducing the size of individual keys. The idea was to show a reduced number of key-rows at a time and allow the user to scroll from one row to another to get the row

with the target key visible. Keyboards with one and two visible rows were evaluated along with a full QWERTY keyboard that had all three rows visible. All three layouts as presented in their study are shown in Figure 1.
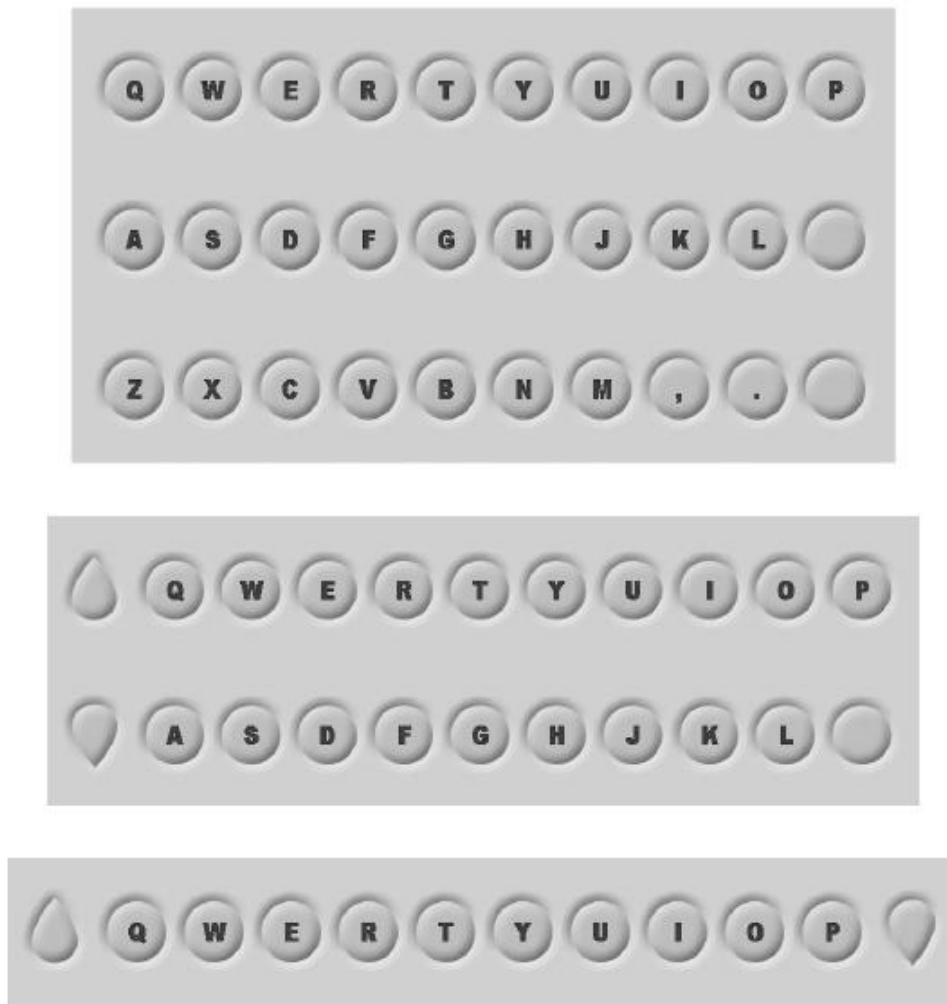

Figure 1. Keyboard Layouts proposed as illustrated in study by Špakov and Majaranta [9]

Text entry speed of all three layouts was evaluated in a user test using gaze pointing and a dwell time of 500 ms. The results with eight able bodied users showed an average speed of 7.26 wpm (STD = 0.95), 11.17 wpm (STD = 1.43) and 14.95 wpm (STD = 1.16) for the 1-row, 2-row and 3-row keyboard, respectively. The usage of scroll keys was also recorded during the test. Scrollable keyboard layout is effective in saving screen space but typing with this keyboard requires additional key presses that may reduce the typing speed.

It was considered worthwhile to improve the layout of the reduced 1-row and 2-row keyboards in order to reduce the usage of scroll keys. With this target in mind, an optimized layout was proposed based on letter-to-letter probabilities. The complete 3-row optimized layout is shown in Figure 2. Eight able bodied users tested the improved layout and showed an average typing speed of 8.86 wpm (STD = 1.70) and 12.18 wpm (STD = 1.99) using gaze control for the 1-row and 2-row keyboard, respectively. Usage of scroll keys was reduced by 18% with the 1-row keyboard and by 40% with the 2-row keyboard. At the same time, the optimized layout added extra complexity of learning an unfamiliar arrangement of keys.
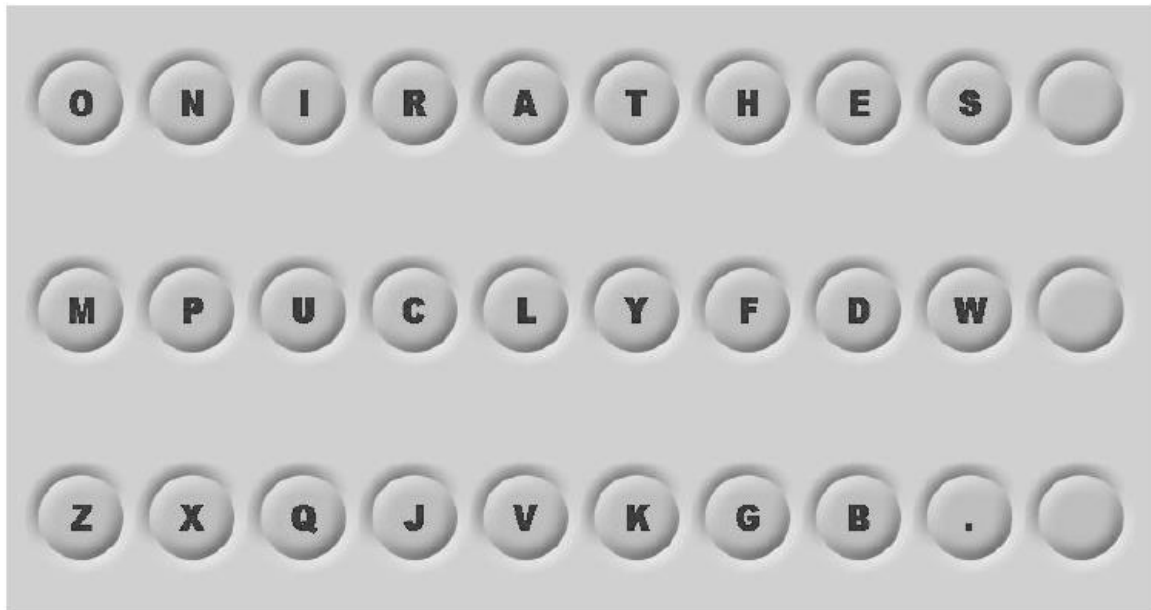

Figure 2. Improved layout of scrollable keyboard proposed by Špakov and Majaranta [9]

Gizatdinova et al. [6] further studied the scrollable virtual keyboard proposed by Špakov and Majaranta. They proposed a vision based perceptual interface. In addition to face/head tracking as a pointing mechanism, the proposed solution used the detection of different visual gestures to operate a scrollable virtual keyboard. Different face gestures were used to issue scrolling and key selection commands. Use of face gestures to scroll key rows eliminated the need for extra key presses. In their evaluation experiments, actual text entry was not performed. A prototype software was implemented to imitate 1 row and 3 row layouts of a scrollable keyboard. Measurement parameters recorded during the experiment were "*task completion time*, *target entry count* (defined as one plus a number

of pointer re-entries to a target within a trial), *complete pointing time* (time interval from the target onset until the last target entry) and *selection time* (time interval from the last target entry event until the selection event)" [6]. Based on the average task completion times, an estimated text entry speed of ~4 wpm was reported for this system. Noteworthy, the results showed that complete pointing time and task completion time increased as the distance "D" between the preceding and current target (i.e. key) increased. This is in accordance with the general assumptions made in the present thesis. However, it is interesting to note that target entry count also increased along with an increase of distance D, indicating a possibly increased difficulty in pointing at the distant keys. Another interesting finding in this work was that with head tracking, it was more convenient to control the cursor movement in horizontal direction than in vertical direction.

In another study, Gizatdinova et al. [5] investigated performance characteristics and subjective satisfaction in text entry using different video based pointing and selection techniques. In their evaluation experiments they compared eye tracking against head tracking as pointing techniques. A static QUERTY keyboard layout was used in their experiment. The experiment with 15 able bodied users showed that gaze pointing resulted in faster text entry speed (M = 10.98 wpm with SD = 0.39) compared to head pointing (M = 4.42 wpm with SD = 0.06) when pointing was combined with character selection by a physical key press. However, text entry with gaze pointing resulted in more errors (M = 8% with SD = 1.77) compared to head pointing (M = 3.8% with SD = 0.3). As was mentioned in the future work chapter of their study, improving robustness and speed of computer vision may improve the text entry speed using head pointing technique.

The experiment results further revealed that the size of the virtual keyboard have a significant impact (faster typing speed with larger size) when using gaze pointing. However, this was not the case with head pointing, where a change in the keyboard size did not have any significant impact on typing speed. In fact, the latest results [Gizatdinova et al., unpublished] revealed that head pointing allows to point at extremely small targets that can be as small as 20x20 pixels. This was probably because the head tracking system used different scaling factor in mapping tracked head position to the cursor movement for keyboards with different key sizes. Thus, the user needed to make the head movement of the same magnitude and required the same level of control over the movement

regardless of the key size. This flexibility of using different scale factor in mapping tracked input to cursor movement would not be feasible with gaze control where cursor has to move exactly with the point-of-gaze. When head pointing was combined with face expression of 'mouth open' and 'brows up', the text entry speed was 3.07 wpm (SD = 0.30) and 2.85 wpm (SD = 0.43) respectively. The brows gesture generated more errors (M = 21% with SD = 15) compared to the mouth gesture (M = 6% with SD = 3). In self-reported subjective feedback, test participants preferred gaze pointing in terms of being more pleasant, faster, more efficient and generally better technique.

Betke et al. [4] implemented a "Camera Mouse" by tracking the motion of a selected body part of the user from video frames to control the movement of a pointer on the screen. In the performance evaluation experiments they used a spelling board application controlled via camera mouse to enter text. The keyboard layout that they used was an alphabetical layout as shown in Figure 3. In their study report they did not describe any rational for their choice of this particular keyboard layout. There was also no performance comparison against any other layout. The average typing speed for 20 able bodied users typing text with this keyboard layout using camera mouse was noted to be 5.86 wpm during the first attempt. An improved average speed of 6.66 wpm was achieved when the same phrase "BOSTON COLLEGE" was typed the third time. Key selection was done by a dwell time of 500 ms during these experiments.
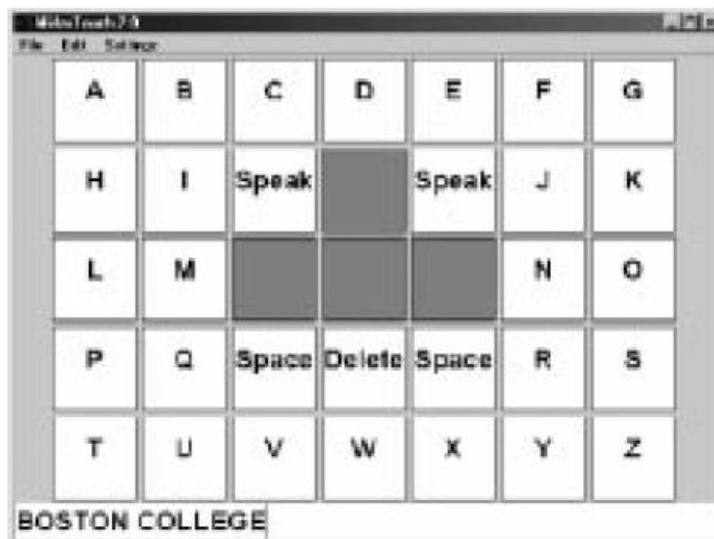

Figure 3. Spelling board used by Batke et al. [4]

Cloud et al. [15] conducted evaluation experiments to further test the camera mouse implemented by Betke et al [4]. Their main focus was to compare the performance of camera mouse using different tracking features. In their experiments, users used camera mouse with three different applications including "Staggered Speech" [16]. Staggered speech uses a two level keyboard system. At the first level, all letters are presented in five groups as shown in Figure 4. Upon selecting a group, each letter from this group is presented at the second level, as shown in Figure 5. Thus, typing each letter requires the user to perform two clicks. This makes it less efficient and laborious to type text with this keyboard. However, the primary objective of this keyboard layout may not be the speed but accuracy for disabled people. As noted on Staggered Speech website that "the program is designed so that buttons don't fall in the same space in successive screens". This avoids unintentional selection of a key when moving from one screen to the next. The average typing speed for 10 able bodied users typing text with this keyboard layout using camera mouse was noted to be 1.85 wpm, 1.63 wpm, 1.67 wpm and 1.52 wpm for nose, lips, eyes and thumb used as a tracking feature, respectively.
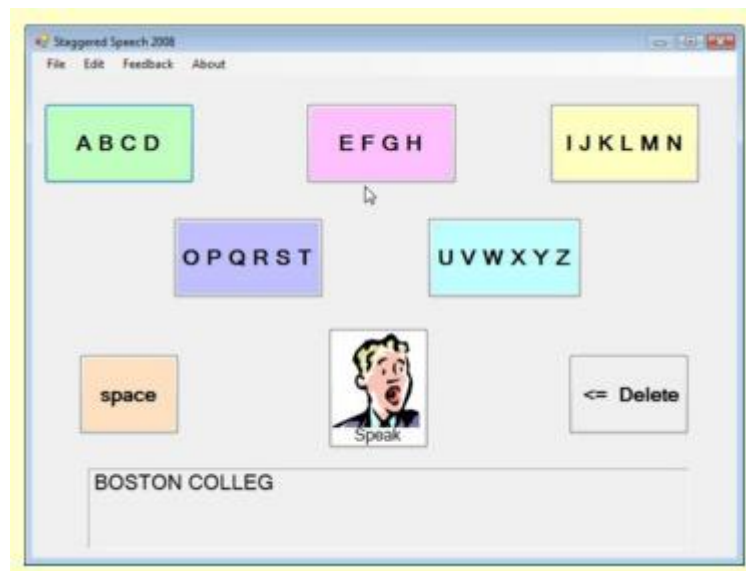

Figure 4. Staggered Speech Keyboard first level [16]
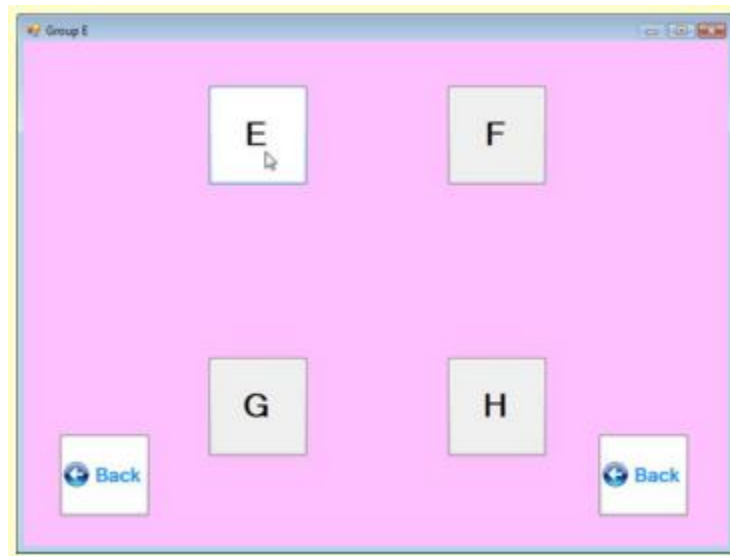
Figure 5. Staggered Speech Keyboard second level [16]

Perini et al. [7] also used computer vision to implement a "face-mouse". They tracked a feature on user's face to control the cursor. Unlike usual computer mouse operation where tracked input is used to decide the exact location of the cursor, they used the tracked input to control the direction in which cursor shall move (aka joystick). In their evaluation experiments, they used a virtual keyboard as shown in Figure 6. The average typing speed for 10 tetraplegic users was 2.7 wpm.

A		B		F		G
	E		J		I	
D		C		L		H
	O	<	A	•	Y	
M		N		R		S
	K		Z		U	
Q		P	W	V		T

Figure 6. Virtual keyboard layout used by Perini et al. [7]

Based on initial results, they improved the layout by adding a dynamic row in the middle. With each entered character, the system predicted the most probable letters. Five of the

most probable next letters were given in the middle row for their easier access, requiring least cursor movement.

A		B		F		G
	E		<		I	
D		C	•	L		H
•	A	E	I	O	U	•
M		N	Y	R		S
	O		Z		U	
Q	W	P	K	V	J	T

Figure 7. Virtual keyboard layout with dynamic character prediction [7]

Another spelling application called GazeTalk, a predictive text entry system developed by Hansen et al. [8] used a dynamic keyboard layout shown in Figure 8. It mainly relies on the word and character prediction that is the main strength of this layout. Originally, GazeTalk was developed to be used with eye tracking input that explains rather large keys of the layout. Hansen et al. compared the performance of GazeTalk when using gaze tracking, head tracking and hand/mouse in a set of user experiments performed during two days. A dwell time of 500 ms was used for key press. The reported text entry speed of 12 able bodied participants with GazeTalk keyboard when used with head tracking was 4.9 wpm (day 1) and 6.10 wpm (day 2). The error rate was 0.5% for typing with head tracking. Unlike the results from Gizatdinova et al. [5], the results from this study did not show any significant text entry speed difference between gaze pointing and head pointing techniques. Subjective ratings of *satisfaction* and *efficiency* also did not show any significant difference between gaze and head pointing techniques in this case. As indicated earlier, results from different studies in this area are not directly comparable due to several differences in study setup, for example the layout of the used keyboard.

This is the text f_		A to Z	Backspace
[8 most likely words]	A	I	O
Space	R	L	U

Figure 8 GazeTalk keyboard layout [8]

Dasher is a non-conventional method of text entry in computing devices. Unlike traditional keyboards with point and select method to enter a desired character, with Dasher, user "navigates" desired characters from a stream of characters by pointing in a two dimensional space. A maximum text entry speed reported with Dasher is 34 wpm using traditional pointing device [17]. De Silva et al. [18] used computer vision (face tracking) to control the pointing at the characters in Dasher. They reported an average text entry speed of 7.3 wpm for two able bodied users. The maximum speed was recorded to be 12 wpm with experienced Dasher user. Dasher uses word prediction based on language model, therefore, text entry speed results may vary based on the prediction hit-rate.
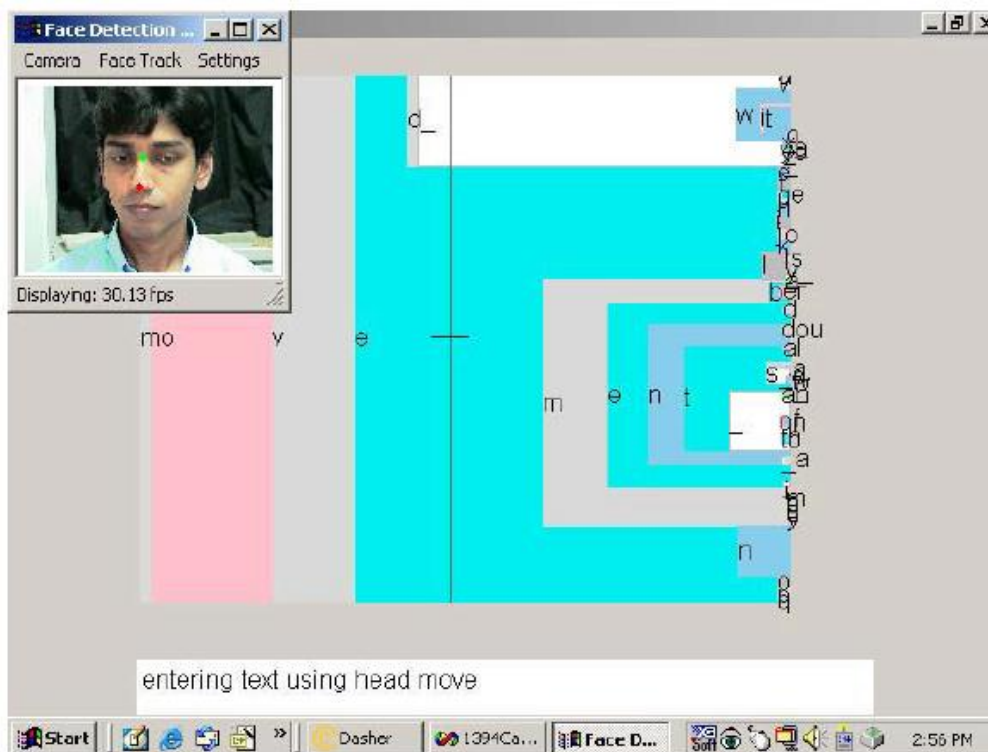
Figure 9 Dasher with face tracking [18]

To conclude, the literature analysis in this chapter reviewed the main prior studies in the field of face typing. A number of empirical investigations confirmed that face/head input can enhance or even completely substitute gaze or hand modality in text entry applications. In general, the speed of face typing was significantly slower than that of eye typing when both conditions were tested in similar experimental setups (with the exception of the study by Hansen et al. [8]). The correctness of the text was rather rarely investigated in face typing studies. It was revealed that face typing resulted in twice less erroneous text than eye typing.

Next, the literature review showed that the majority of the past work used static QWERTY or alphabetic, keyboard layouts [5] [9]. Even in those cases when some optimized dynamic layouts were introduced, for example, in studies [7] [8] [15], the authors did not reveal why certain design solutions were chosen neither it was clear what improvements the design solutions brought to the overall typing performance and experience of the users. The field needs more research to investigate usability factors to be taken into consideration in designing keyboard layouts for face typing systems. There

is a room for further study to explore alternate layouts and to study their impact on the speed and accuracy of text entry as well as to make it easier and less stressing, both mentally and physically, for the user.

When it comes to the user experience and subjective evaluation of hands-free text entry solutions, it is even harder to compare the results presented in different studies. These results differ not only because of the differences in test setups but also due to differences in the user characteristics. It can be noticed from previous studies that the user experience varies between abled users and users with disability or physical limitations. Betke et al. [4] reported quite positive subjective feedback about *the camera mouse* from the users with cerebral palsy and traumatic brain injury. In other studies, for example [8], able bodied users often felt that hands-free text entry solutions need to improve on speed, accuracy and ease-of-use as they tend to compare the performance against the systems for typing with hands that they use regularly. Although, users with disabilities are often considered primary users of hands-free typing systems, feedback from abled bodied users is equally important. Such users can directly compare the performance of the solution-under-study with the default solutions they use in their everyday life. This should help in reducing the usability gaps between hands-free text entry systems and traditional systems of text entry with hands.

3. Design

The design of a new layout of the virtual keyboard proposed in this thesis was inspired by the Fitts' law [10] [11][ISO 9241-9:2000].

3.1 The Fitt's Law

The original Fitts' law [10] explains the speed and accuracy tradeoff of a pointing device. According to it, it is more difficult to point at the targets that are smaller and located farther away:

$$MT = a + b \, log2 \, (2A/W) \quad Eq. \, 1$$

Where MT is the movement time, a and b are constants, A is the distance to the target and W is the width of the target. From Equation 1, it can be seen that the movement time is directly proportional to the target distance and inversely proportional to the width of the target. The variable part of the Fitts' law equation is termed as *index of difficulty* (ID) and is represented as

$$ID = log_2 \, (2A/W) \qquad Eq. \, 2$$

The later formulations of the Fitts' law [11][ISO 9241-9:2000] take into account the accuracy measure - the rate of trials where selection did not hit the target. This information can be captured by analysis of SD_x standard deviation in x-coordinate recorded over a block of trials. Then target width W can be substituted by effective target width W_e:

$$W_e = 4.133 \, SD_x \qquad Eq. \, 3$$

$$ID_e = log_2 \, (A/W_e+1) \qquad Eq. \, 4$$

Nevertheless, the relationship between the width of the target and the pointing distance remains essentially the same.

3.2   Design of the dynamic keyboard

Based on the Fitts' law, one feels encouraged to increase the size of UI elements to decrease the difficulty of their access. This has been routinely done in eye typing where the size of the keyboard typically occupies the biggest part of the screen in order to compensate for inherit inaccuracy of eye tracking devices [9] [8]. Eye movements are very fast and the increase of the keyboard size does not affect negatively the overall speed of typing. When it comes to face typing, the pointing is implemented by head motion that may be more demanding as compared to eye movements in terms of physical effort. The increased size of UI elements implies longer distances between the keys in a layout, longer travel times for the head pointer and, as a result, stronger head movements to be performed by the user which may be not always feasible for the user. It is possible to use gain factors to "amplify" head movements so that smaller head movements will result in stronger "jumps" of the head pointer [5]. However, this may lead to a decreased accuracy of pointing at the elements of the interface. That is, the starting point for the design of the new layout was to enlarge some elements of the layout while keeping the total size of the keyboard relatively small/unchanged. Following the definition of the index of difficulty, the keys nearby the last pressed key are easier to access whereas the farther keys are harder to access. It can be expected that giving larger size to the farther keys should lower the level of difficulty in accessing these keys and, at the end, improve the keyboard performance. Based on this idea, a novice design of the virtual keyboard was proposed.

In a typical QWERTY keyboard, each key is given equal fixed size as shown in Figure 10. Such keyboard is termed as *the static layout* in this document. The novice design of the keyboard layout proposed in this thesis work is termed as *the dynamic layout*. In the dynamic layout, the basic arrangement of the keys is the same as in traditional static layout. However, unlike static keyboard, the dynamic layout adjusts the size of the keys dynamically as the user enters text. Each time the user presses a key, the size of each key is recalculated "on the fly" based on the key distance from the pressed key: the longer the distance, the larger the size of the key. The overall size of the keyboard remains unchanged - the space available within the overall keyboard area is redistributed among all keys of the layout.

The apparatus used in the evaluation experiment is described in details later in this document. To make it easier to understand the implementation details of the two layouts, some of the numbers from the used apparatus are used here as example. The overall size of the keyboard was set to cover 80% of the available desktop width. On a system with the screen resolution of 1280x1024 pixels (17 inch display), the overall size of the keyboard was 1024 pixels in width. There were 11 keys in each of the 3 rows of the keyboard. This gave each key a size of 93x93 pixels on the static layout. This size is referred as the *nominal* key size in next paragraph when describing the details of calculating key size for the dynamic layout. On the dynamic layout, the smallest key size was 60x68 pixels and the largest key size was 119x102 pixels.

Implementation of the keyboard was done in Qt framework using C++ programming language. Like many other graphical user interface frameworks, it also offers a readily available push button class called QPushButton. However, this class as-it-is was not very useful to implement the keys of the proposed dynamic layout. Target was to utilize the full space available within the keyboard as well as to have a smooth change of key size without having a *staircase* look to get an aesthetically pleasing keyboard design. To achieve this, a custom graphical user interface element was implemented as an overlay where each key was a polygon with four vertices: the four corners of a quadrilateral key. This solution enabled setting each vertex of the polygon freely to achieve the desired variation in key size. Location of each vertex was calculated as a sum of a fix part (Fix) and a variable part (Var).

$$Vert = Fix + Var \qquad\qquad Eq.\ 4$$

Fix part was calculated based on a self-defined the smallest-allowed-size of the key. In this implementation, the smallest allowed size was set to 60 pixels. Variable part was directly proportional to the distance (D) of this vertex from the center of the last pressed key. In order to fulfill the target that the keys with recalculated sizes must cumulatively be an exact-fit to the whole keyboard area, the space (S) between the last pressed key and the keyboard boundary in the direction of this vertex was also used as a factor in the calculation. In practice, a trial-and-error approach was used to reach an implementation that satisfied the required behavior. The final implementation can be described mathematically as following;

$$Var = \frac{K * D^2}{S} \qquad\qquad Eq.\ 5$$

Here K is a constant that is derived based on the size delta of the smallest allowed key size in dynamic layout (60 pixels) and the *nominal* key size. It is important to note that the equations 4 & 5 do not calculate the size of a key but the location of the vertices to achieve the intended key size.

The implementation of the dynamic layout had an extra latency over the static counterpart: upon every key-press, size of each key was to be re-calculated and each key was to be re-drawn in the software. Software trace log shows that this extra processing took around 52 ms time upon each key-press. In other words, the dynamic layout had an extra latency of 52 ms over the static layout for every keystroke. One could consider to improve the software implementation to reduce this latency. However, the delay was so small that the rearrangement of the keys in the layout can be considered real-time. Text entry speed results given in results chapter confirmed that this additional latency did not have a significant impact on the text entry speed.

Figure 11 shows the dynamic layout after the user has pressed key 'H'. In this view, key 'H' has the smallest size. All other keys have a gradually increasing size as a function of their distance from the last pressed key 'H'. Similarly, Figure 12 shows the dynamic layout view when the user has pressed the key 'A'.

Q	W	E	R	T +	Y	U	I	O	P	<-
A	S	D	F	G	H	J	K	L	Ä	Å
Z	X	C	V		B	N	M	Ö	,	.

Figure 10. Static QWERTY layout keyboard with letters of Finnish alphabet, punctuation characters, and BACKSPACE and SPACE function keys. A cross and dark-grey color identify the key "T" that is highlighted by a pointer.

Q	W	E	R	T	Y	U	I	O	P	<-
A	S	D	F	G	H +	J	K	L	Ä	Å
Z	X	C	V		B	N	M	Ö	,	.

Figure 11. Dynamic QWERTY layout with 'H' key pressed.

Q	W	E	R	T	Y	U	I	O	P	<-
A +	S	D	F	G	H	J	K	L	Ä	Å
Z	X	C	V		B	N	M	Ö	,	.

Figure 12. Dynamic QWERTY layout with 'A' key pressed.

As it can be seen from Figure 12, the size of the last pressed key ('A' in this case) is the smallest and width of the most horizontally distant keys (BACKSPACE, 'Å' or "dot" key) are the largest. Similarly, the height of any key is also increased based on its distance from the last pressed key. Font size of the key labels is dependent on the key size, hence different keys have different sized labels. Every time a key is pressed, the keyboard layout is updated by adjusting the size of the keys based on new distance values. The cursor is moved to the center of the pressed key in order to keep it within the key boundary. Idea is that if a character is to be entered twice then the user must be able to re-enter it without needing to move the cursor. This is always the case with the static layout therefore the dynamic layout is also made to have the same behavior.

4. User Study

The aim of the user study was to evaluate usability characteristics of the proposed dynamic layout of a virtual keyboard for hands-free text typing with computer vision based head pointing. In the evaluation, text entry tasks were performed with the dynamic and static layouts and a comparison was made to identify if there was any difference in text entry speed, accuracy and user satisfaction between the two layouts. The usability characteristics were examined as suggested in [5]: (1) *typing speed* as a time required to type a target phrase in words per minute (wpm), (2) *relative error rate* calculated using Levenshtein string distance algorithm [19] as a ratio of erroneous or missed characters to the total number of characters in a phrase, (3) *keystrokes per character* that defines how many times a user presses a key to enter a single character, in this case, indicating the number of corrections and (4) *self-reported subjective evaluation* of user experiences. Therefore, in addition to measuring the qualitative characteristics of text entry, participants were also asked to provide subjective feedback by filling in the evaluation scales.

4.1 Participants

Sixteen students of between 18 and 50 years of age (M = 26.87, SD = 8.08) from local university volunteered to participate in the evaluation experiment. Thirteen were male and three were female. Based on the participants' self-report, six participants had normal vision, seven participants were wearing eye glasses and three were wearing contact lenses. Two participants claimed that they had some experience of computer vision and hands free text typing. One participant claimed to have some experience of computer vision but not of hands free text typing. One participant claimed to have some experience of hands free text typing but not of computer vision. Other participants had no previous experience of either computer vision or hands free text entry. All participants had good experience of using computers for leisure and study/research purposes and therefore had good experience of typing text with static QWERTY keyboard.

4.2   Experiment design

Experiment was a within-subject factorial design with the independent variable being the layout of the keyboard (with two levels: dynamic and static layout) and the dependent variables were typing speed, relative error rate, keystrokes per character and subjective evaluation. Each experiment had two blocks, in which the participants typed text with the dynamic and the static layouts. In each block the participants typed eight phrases. The experiment was counterbalanced regarding the layout in a way that half of the participants started typing text with the dynamic layout in the first block. The total number of phrases typed was 256 phrases (16 participants x 2 layouts x 8 phrases).

4.3   Apparatus

Experiments were run in the gaze lab of the University of Tampere. The illumination conditions were kept the same in every experiment.

The corpus of 500 English phrases was compiled by MacKenzie [20]. Each phrase was roughly about 30 characters (including spaces and punctuation marks) long.

The hardware and software used during the experiment was as follows. A computer with 2.66 gigahertz (GHz) Intel Core 2 Quad core processing unit (CPU), 4 gigabyte (GB) random access memory (RAM) and a GeForce 9600 GT display adapter. 17-inch display was set to 1280x1024 pixels resolution. The computer was running a 64 bit Windows 7 Service Pack 1 operating system (OS). A Logitech HD Pro Webcam C920 camera with 25 frames per second capture rate and 1920x1080 pixel resolution was used to get video stream of the participants' head to perform head tracking. Distance between the user's head and the computer display was about 50 cm.

A text entry software prototype *myKeyboard* was implemented in Qt framework using C++ programming language to evaluate text entry performances of the dynamic and the static keyboard layouts using head tracker as the pointing mechanism. The graphical user interface (GUI) of the prototype contained a text label showing the phrase to be typed and a text box showing the typed text, both in capital letters. Figure 13 shows a screen shot of the prototype with the static keyboard layout selected. Whereas Figure 14 shows a screen shot of the prototype with the dynamic keyboard layout selected. When the

keyboard with the static layout was presented to the participant for the first time, cursor was positioned in the middle of the keyboard over key 'H' as shown in Figure 13. When the keyboard with the dynamic layout was presented for the first time, it was laid-out with key 'H' as the last pressed key and cursor was positioned in the middle of it. Therefore, key 'H' had the smallest key size and all other keys were sized according to their distance from key 'H' as shown in Figure 14.

Figure 13. Text entry prototype "myKeyboard" with the static keyboard option selected.
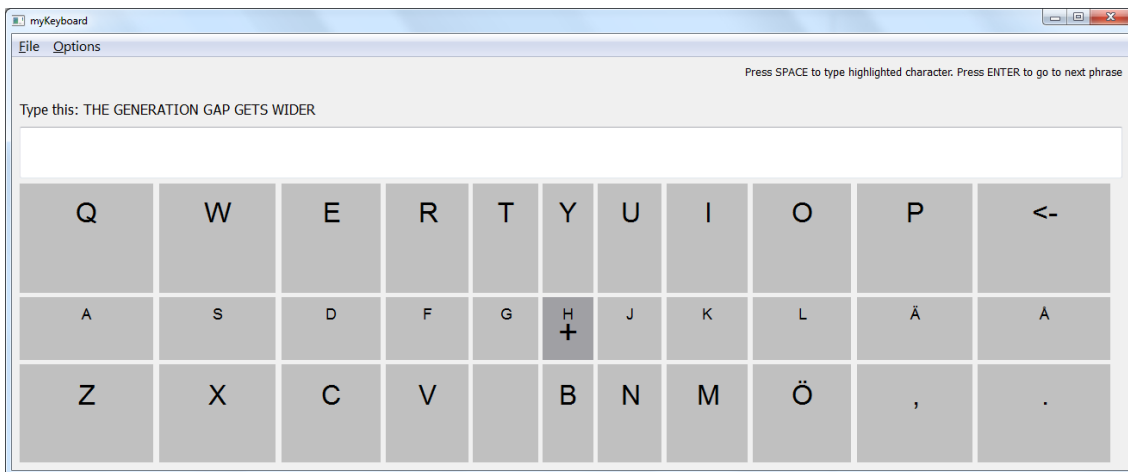

Figure 14. Text entry prototype "myKeyboard" with dynamic keyboard option selected.

An existing head/face tracking system called *Fanalyzer* implemented in Qt framework was used to steer the cursor movement [5] [6]. The participant observed the output of *Fanalyzer* as a visual feedback so that the participant could adjust his/her position in front of the camera, as Figure 15 shows.
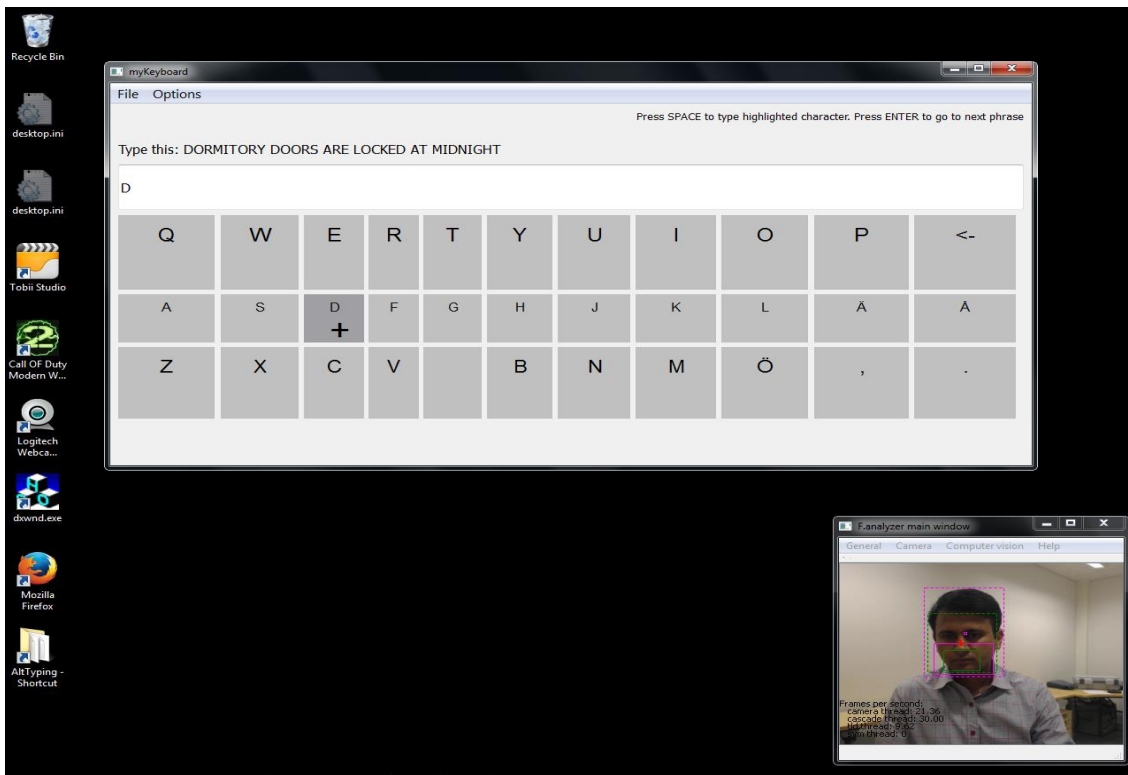
Figure 15. Computer screen view during an experiment block.
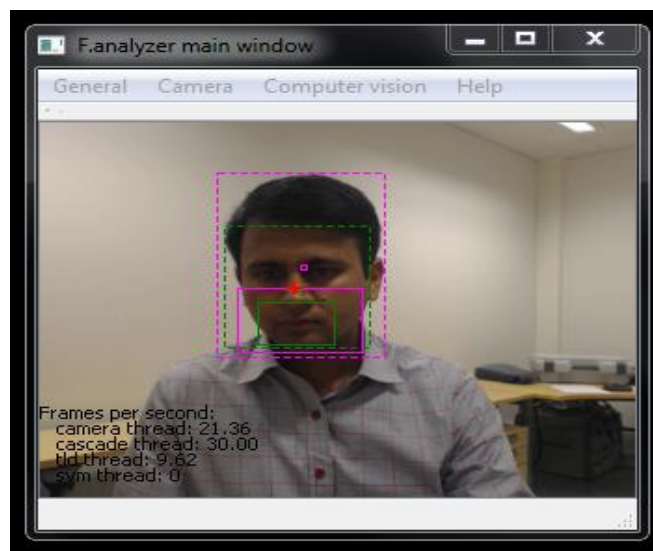
Figure 16. Output of Fanalyzer, showing tracked face and mouth area with violet and green rectangles.

Fanalyzer output window is shown in Figure 16. In this figure, the green coloured rectangle indicates the tracked mouth area which is used for mouth expression classifying. This output was not used by *myKeyboard*. The violet coloured rectangle over the face area shows the tracked face area. Violet coloured small circle indicates the centre point of this rectangle. Location of this centre point was used as input to control the cursor

for the keyboard. Application *myKeyboard* worked as a client of face tracking service *Fanalyzer* that was running as a server. *Fanalyzer* tracked the coordinates of the user's face and sent those to *myKeyboard* application. The received coordinates were averaged over 6 input values to get a smoother cursor movement. Once the cursor was above the target key, a key-press was performed by pressing the *space-bar* key of the physical keyboard of the computer system. *Enter* key from physical keyboard was used to move to the next phrase. A beep sound was played as a feedback to the user for each entered key.

4.4   Procedure

In the beginning of the experiment, the participants were familiarized with the focus and objectives of the research. The background information form and informed consent form were filled by the participants.

The participants were instructed to sit comfortably in front of the screen, but also in a way that they could move their head in all directions. Participants were advised to avoid head rotation and preferably move torso to do pointing by head as suggested in [5]. Text typing application *myKeyboard* was placed in the top half of the screen and the face tracker *Fanalyzer* window was placed below it in the bottom-right corner of the screen, as shown in Figure 15. The camera settings were carefully adjusted so that the head of the participant occupied the same area of the camera image as shown in Figure 15. This was to ensure that the participants' head remained in the camera view all the time, even when the participants performed head motions. *Fanalyzer* and its output visualization was explained to the participants.

Then the face tracker calibration procedure was performed as proposed earlier for this system [5]. In this calibration process, the participants were required to control the movement of an on screen pointer by moving their head accessing four corners of a calibration window, as Figure 17 illustrates. In this figure, the read coloured block indicates the block currently accessed with the cursor shown as blue coloured cross mark. During this process, *Fanalyzer* collected image data and trained the face tracker algorithm.

Figure 17. Fanalyzer Face Tracker Calibration Window.

After the calibration of the face tracker, participants were familiarized with the text typing application *myKeyboard*. Participants were told about different UI elements and controls to enter the text. Then participants were allowed to do a little practice task by typing own name once with each layout using head tracking. In case of all participants except one, calibrating the system once was enough to start typing text. For one participant the system had to be re-calibrated additionally once.

After the practice trial, the actual experimental trial started in which participants typed eight randomly selected phrases from the phrase corpus using the first layout. Participants were told to type as fast as they could but also as accurate as they could. Error corrections were allowed but not required. After this, the participants typed text with another layout. The participants were given a chance to rest between the two blocks of the experiment.

At the end of each block, participants were told to fill in a subjective scale questionnaire to rate the used layout using the evaluation scales of *general evaluation, pleasantness, dominance, quickness, accuracy, efficiency, tiredness, distractibility and mental effort* [5]. The questionnaire is given in Appendix 1. The rating scales were so called bipolar rating scales meaning that each end of the scale represented opposite ends of a two way continuum. The scales had nine points varying from -4 to +4. The rating scales varied from negative (left negative side) to positive (right positive side) evaluation, for example, the more positive the value, the better the participant's experience was. The middle area of each continuum represented a neutral point in between the opposites. The scales were explained to the participants.

At the end of the experiment, the participants were asked to fill in a comparative questionnaire to indicate their personal preference among the two layouts for *general evaluation, pleasantness, dominance, quickness, accuracy, efficiency, tiredness, distractibility, mental effort and overall preference.* The questionnaire is given in Appendix 2.

The target phrase, the actual text typed, the time taken to type each phrase, the error count and the keystrokes to type each phrase were recorded into a log file. The experiment took approximately between 30 to 60 minutes.

5. Results

5.1 Data pre-processing, outlier removal and analysis methods

Data recorded during the experiment was analyzed to compare the text entry performance of two keyboard layouts. Among objective data, the performance was analyzed in terms of text entry speed, relative error rate and keystrokes per character (KSPC). Text entry speed was recorded starting from the moment when a phrase was presented to the participant until the last character was entered before the participant moved to next phrase. Text entry speed was then calculated by dividing the phrase length in words (5 characters) by the time duration in minutes;

$$WPM = Phrase\_length_{words} / Typing\_duration_{minutes} \qquad Eq.\ 6$$

Relative error rate was calculated based on *Levenshtein string distance algorithm* [18] as a proportion of incorrect or missed characters to a total number of characters;

$$Rele\ Error = (Levenshtein\_error_{characters} / Phrase\_length_{characters}) * 100 \qquad Eq.\ 7$$

KSPC was calculated by recording the number of total key-presses done during typing of a phrase and then dividing this number by the number of characters in the typed phrase;

$$KSPC = Phrase\_length_{characters} / Key\text{-}press_{count} \qquad Eq.\ 8$$

Objective data gathered from each test participant was first iteratively analyzed for outliers one data point at a time according to the exclusion criteria as follows. At each step, the mean and standard deviation were computed over the data of a particular participant. Then the largest and smallest values with a delta larger than 2 standard deviations from the mean value for this participant were considered as outliers and removed from the analysis. This process continued until the data of each participant was free of extreme deviations from the mean value.

As a result, two out of 128 speed data records for static layout and three out of 128 speed data records for dynamic layout were dropped out from the analysis as extreme outliers. Six out of 128 error rate data entries for each (static and dynamic) layout were dropped as outliers. Five out of 128 KSPC data records for each layout were dropped out as outlier entries.

Finally, grand mean, standard deviation (SD) and standard error of means (SEM) were computed over the outlier-free mean values of all participant data. Student's t-test for two-tailed distributed paired data was used to analyze objective data.

In addition to objective data, self-reported subjective feedback of test participants was analyzed using Wilcoxon signed ranks test for two-tailed paired samples. A standard alpha of 0.05 was defined as a significance level.

5.2 Text Entry Speed

The grand mean text entry speed of all test participants was 5.14 wpm (SD = 1.06, SEM = 0.11) with static layout and 5.03 wpm (SD = 1.17, SEM = 0.11) with dynamic layout. T-test did not reveal significant differences in text entry speed between the layouts, $p\_calculated = 0.59$, $p\_calculated > 0.05$. As noted earlier, the current implementation of the dynamic keyboard had an extra latency of 52 ms in rearranging the keys in the layout. After excluding this extra latency from the measured values, the grand mean text entry speed of dynamic layout was 5.15 wpm (SD = 1.23, SEM = 0.12). T-test did not reveal significant differences in this case either, $p\_calculated = 0.97$, $p\_calculated > 0.05$. Therefore this additional latency was considered ignorable and was not studied any further.
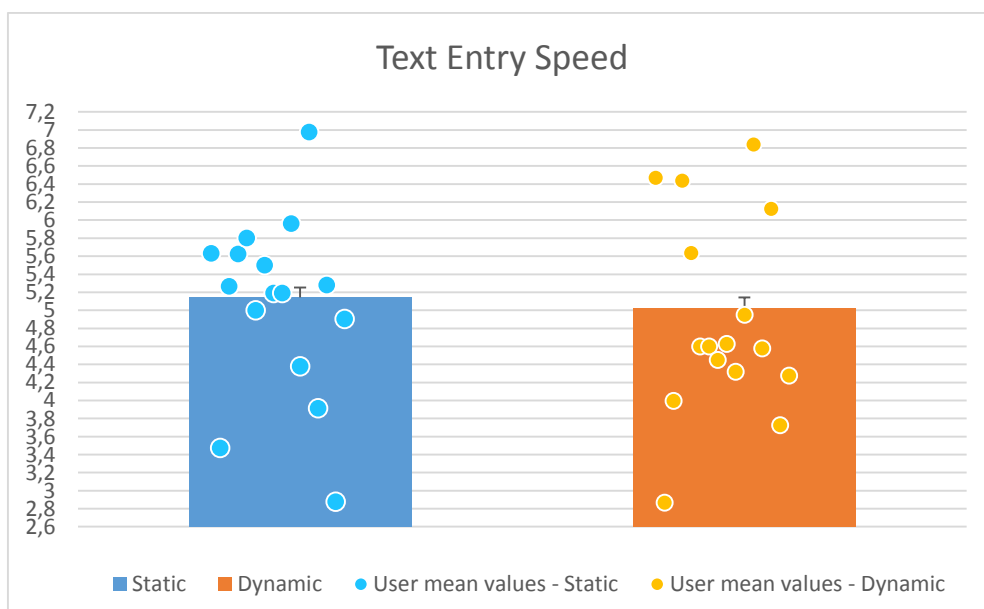
Figure 18. Text entry speed: Words Per Minute (WPM) with Standard Error of Means (SEM), outliers removed. Mean text entry speed values of each user with both layouts are also shown as vertical axis of scatter chart.

5.3 Relative Error Rate

The grand mean relative error rate was 1.28% (SD = 2.07, SEM = 0.32) with static layout and 0.83% (SD = 1.46, SEM = 0.20) with dynamic layout as shown in Figure 19. T-test did not reveal significant difference in error rate results for the layouts, p_calculated = 0.29, p_calculated > 0.05.
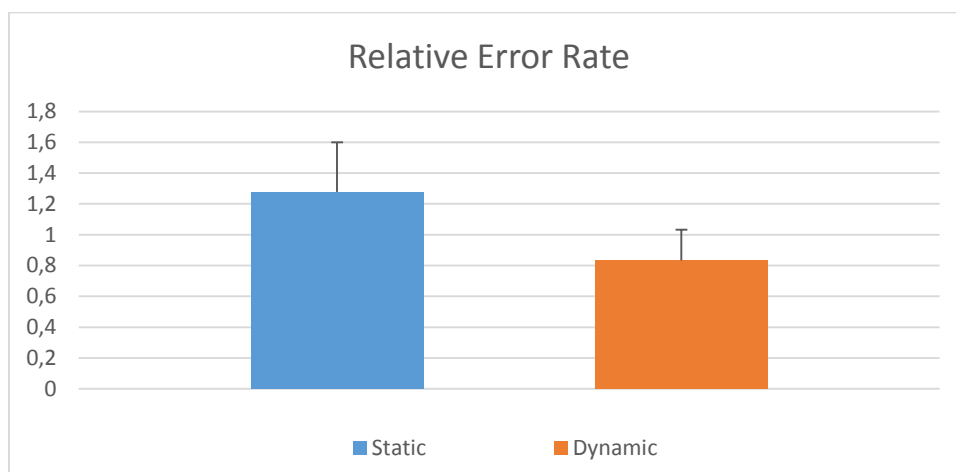
Figure 19. Relative Error Rate with Standard Error of Means (SEM), outliers removed

5.4 Keystrokes per Character

The grand mean value of KSPC was 1.05 (SEM = 0.01) for both layouts with SD = 0.04 for static layout and SD = 0.06 for dynamic layout as shown in Figure 20. T-test did not reveal significant difference in KSPC results for both layouts, p_calculated = 0.58, p_calculated > 0.05.
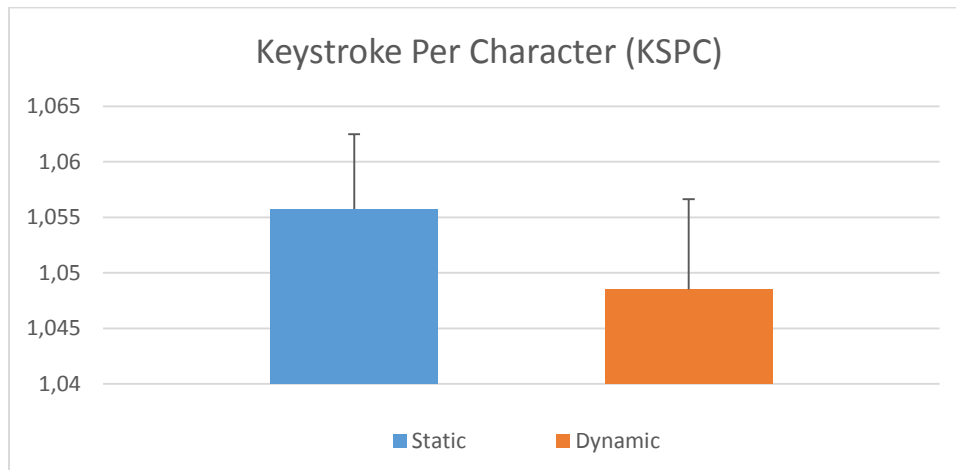
Figure 20. Keystroke per Character with Standard Error of Means (SEM), outliers removed

5.5 Subjective Ratings

The participants considered static layout to be more accurate and requiring less mental effort. All other results found to be statistically insignificant. The participants' self-reported subjective ratings of two layouts are listed in Table 1 and graphically represented in Figure 21.
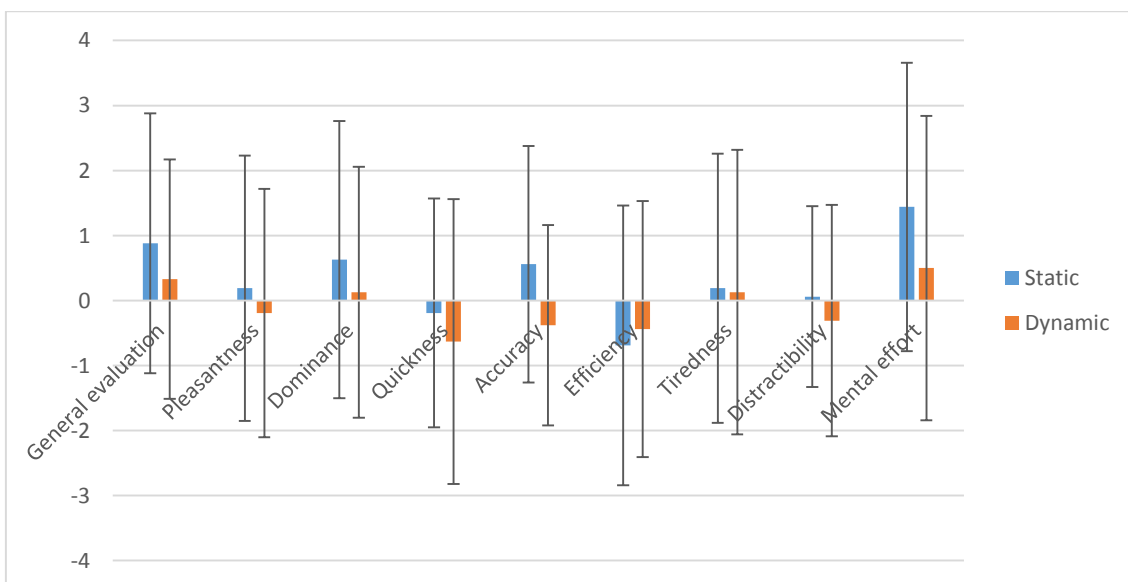
Figure 21. Self-reported subjective ratings, Mean and SD values

Rating	Layout	Mean	SD	Min	Max	Test
General Evaluation	Static	0.88	2	-3	4	Z = 1.15
	Dynamic	0.33	1.84	-4	3	P = 0.12
Pleasantness	Static	0.19	2.04	-3	4	Z = 1.00
	Dynamic	-0.19	1.91	-4	3	P = 0.16
Dominance	Static	0.63	2.13	-3	4	Z = 0.77
	Dynamic	0.13	1.93	-2	4	P = 0.22
Quickness	Static	-0.19	1.76	-3	3	Z = 0.81
	Dynamic	-0.63	2.19	-4	3	P = 0.21
Accuracy	Static	0.56	1.82	-2	3	Z = 2.40
	Dynamic	-0.38	1.54	-3	3	P = 0.01
Efficiency	Static	-0.69	2.15	-4	3	Z = 0.36
	Dynamic	-0.44	1.97	-4	3	P = 0.36
Tiredness	Static	0.19	2.07	-3	4	Z = 0.16
	Dynamic	0.13	2.19	-3	4	P = 0.44
Distractibility	Static	0.06	1.39	-3	3	Z = 0.90
	Dynamic	-0.31	1.78	-3	3	P = 0.18
Mental effort	Static	1.44	2.22	-3	4	Z = 2.70
	Dynamic	0.5	2.34	-3	4	P = 0.00

Table 1. Self-reported subjective ratings

The results of the preference evaluation are presented in Figure 22. Most of the participants preferred the static layout over the dynamic layout. For example, six of the participants considered the dynamic layout to be more distractive while two participants had considered the static layout to be more distractive. Eight participants thought that the dynamic layout required more mental effort while two had the opinion that the static layout required more mental effort. Twelve participants perceived the static layout to be more accurate, only three participants perceived the dynamic layout to be more accurate. Overall, ten participants preferred the static layout, three participants preferred dynamic and three did not have any preference.
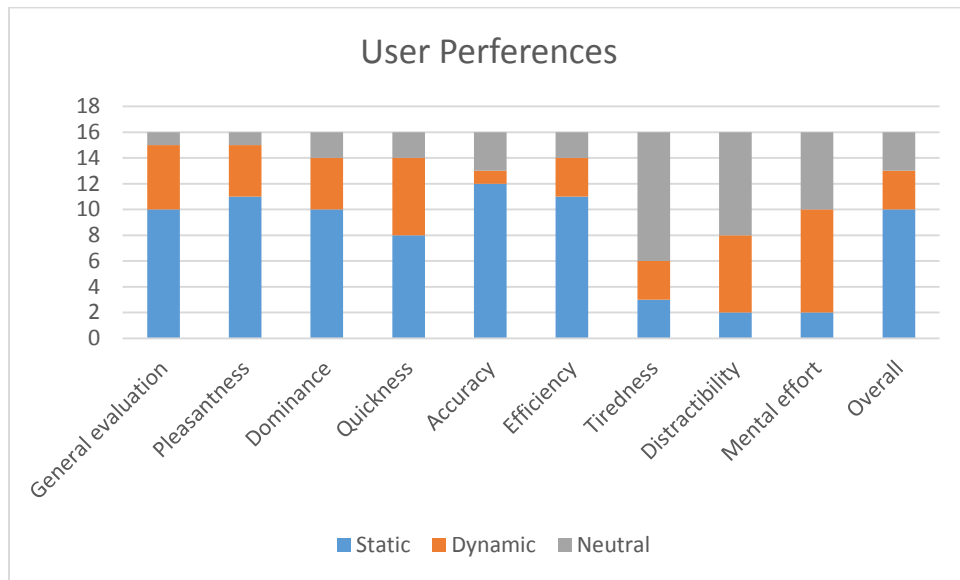
Figure 22. User preference of the two layouts. Y-axis shows the count of preferences given to each of the three options: static, dynamic and no-preference.

6. Discussion

Inspired by the Fitts' law, a new layout design was proposed. In the new design, the size of each key was dynamically calculated based on its distance from the last pressed key. In this experiment, the performance of the newly proposed dynamic layout for hands-free text entry was investigated as compared to the performance of more traditional static layout. The analysis of the results did not show statistically significant differences between objective performance characteristics of the layouts. Thus, in case of the dynamic layout the text entry speed was 5.03 wpm, relative error rate was 0.83% and KSPC was 1.05. In case of the static layout, the text entry speed was 5.14 wpm, relative error rate was 1.28% and KSPC was 1.05.

It is not so straightforward to compare the text entry performance achieved in this study against the results from previous studies on text entry with face/head pointing. Face/head tracking based text entry systems proposed in different studies differ greatly from each other. For example, different studies used different key-press mechanism. Different text entry systems in previous studies required different amount of key presses to enter a single character. Some systems used character and/or word predictions which made the text entry performance with these systems dependent to the success rate of the prediction. Therefore there is no straightforward way to compare the text entry performance of different systems which used face/head tracking. The dynamic design gave equally good text entry speed and accuracy performance compare to some of the previous solutions, for example Gizatdinova et al. [6]. It achieved better text entry speed than some of the previous work, for example Cloud et al. [15], Perini et al. [7] and Hansen et al. [8]. *Dasher* based solution from De Silva et al. [18] had a better text entry speed (M = 7.3 wpm) than that of the current work.

The current results can be directly compared to the earlier results by Gizatdinova et al. [5]. The test in this study were conducted on the same machine in the same lab; the input from a physical keyboard was used as a key-press. They reported a mean text entry speed of 4.42 wpm (SD = 0.39) and error rate of M = 3.8% (SD = 0.3) with head pointing. Thus, the current results confirmed the earlier results and showed comparable means of text

entry speed with head pointing, and also quite small variance (or standard deviation) of text entry speed measurements.

Although the results did not reveal significant improvement of text entry performance with the dynamic layout, it is a fact that the new layout did not deteriorate the typing performance as compared to the traditional static layout. In the following text, several considerations about the proposed design and their influence on the resulting typing performance will be discussed.

In the current implementation of the layout, the smallest key size was 60x68 pixels and the largest key size was 119x102 pixels. As this was a novice design idea, there was not much previous data available to base upon the decision of key size ratio. The idea was to have a reasonable enough size delta between the smallest and the largest key. In this case the largest key was almost double the size of the smallest key. However, it may be that a given increase in the size of the distant keys was not perceived by the participants as large enough to make quick and strong (and less accurate) "jumps" from the last pressed key to a distant key. In other words, it is possible that the participants did not learn to take the advantage of the proposed layout.

Thus, future design improvements can take into consideration regarding the relation between the smallest and the biggest key sizes in the layout to obtain the advantage of easier and faster pointing. The smallest size of the keys was selected based on the background literature analysis [4]. However, new results [Gizatdinova et al., unpublished] appeared after the experimental stage of this thesis has been completed, revealing that head pointing allows to point at extremely small targets that can be as small as 20x20 pixels (in similar conditions). That is, one idea is to trial with more extreme ratio of the smallest and the largest key sizes. The last pressed key on the keyboard can be made even smaller, (for example, 35x35) that may allow to make the distant keys more than 200 pixels large. If distant keys are large enough, the participants may prefer to make fast and rough head movements to move the pointer from the last pressed key to the distant keys rather than to accurately steer the pointer to the desired location. It is possible to argue that in such conditions the typing performance of the proposed dynamic layout may outperform that of static layout.

As mentioned earlier, all the test participants of this user study were experienced computer users. Though most of them did not have any prior experience with face/head tracking based text entry, they were very experienced in using traditional (static) QWERTY keyboard. The dynamic layout, being a novice idea, was a totally new experience for them. This might be the reason why a majority of the test participants perceived dynamic layout to be less accurate and requiring more mental effort in the self-reported subjective feedback. Objective results did not support the user perception about accuracy and showed that the dynamic layout was equally accurate as the static layout. Participants were given very little time to get familiar with the layout. A longer practice session to get familiar with the new layout might have helped the participants to perform better in text typing tasks. Explicit instructions to participants to take advantage of the larger targets and make fast head movements may also improve the text entry performance with the dynamic layout. In practice task, using such phrases that have characters located far apart from each other may also train the participants better to take advantage of the larger targets and make fast and coarse head movements.

7. Conclusion

A novice layout of virtual keyboard optimized for face/head tracking based text entry was proposed. The text entry performance of the proposed layout was tested against the traditional QWERTY keyboard in user test. Objective results of text entry speed, relative error rate and keystrokes per character as well as subjective feedback was recorded and analyzed. The performance difference results were found to be insignificant other than the subjective scores of accuracy and mental effort where static layout was preferred. However, objective results did not show any degradation of text entry performance with using the dynamic layout. A variation of the dynamic layout to have larger ratio of key sizes was proposed as a future study.

References

[1] A. Sears, "Improving Touchscreen Keyboards," *Interacting with Computers,* vol. 3, no. 3, pp. 253-269, 1991.

[2] P. Majaranta and K.-J. Räihä, "Twenty years of eye typing: systems and design issues," in *Proceedings of the 2002 symposium on Eye tracking research & applications, pp. 15-22*, New York, USA, 2002.

[3] R. J. K. Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Transactions on Information Systems (TOIS) - Special issue on computer—human interaction,* vol. 9, no. 2, pp. 152-169 , 1991.

[4] M. Betke,, J. Gips, and P. Flemming, "The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities," *IEEE Transactions on Neural Systems and Rehabilitation Engineering ,* vol. 10, no. 1, pp. 1-10, 2002.

[5] Y. Gizatdinova, O. Špakov and V. Surakka, "Comparison of Video-Based Pointing and Selection Techniques for Hands-Free Text Entry," in *AVI '12 Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 132-139*, New York, USA, 2012.

[6] Y. Gizatdinova, O. Špakov and V. Surakka, "Face Typing: Vision-Based Perceptual Interface for Hands-Free Text Entry with a Scrollable Virtual Keyboard," *IEEE Workshop on the Applications of Computer Vision (WACV'12),* pp. 81-87, 2012.

[7] e. perini, s. soria, a. prati and r. cucchiara, "FaceMouse: A Human-Computer Interface for Tetraplegic People," in *ECCV'06 Proceedings of the 2006 international conference on Computer Vision in Human-Computer Interaction, pp. 99-108*, Berlin, Germany, 2006.

[8] J. P. Hansen, K. Tørning, S. A. Johansen, K. Itoh and H. Aoki, "Gaze Typing Compared with Input by Head and Hand," in *Proceedings of the 2004 symposium on Eye tracking research & applications, pp. 131-138*, San Antonio, TX, USA, 2004.

[9] O. Špakov and P. Majaranta, "Scrollable keyboards for casual eye typing," *PsychNology Journal,* vol. 7, no. 2, p. 159–173, 2009.

[10] P. M. Fitts and J. R. Peterson, "The information capacity of the human motor system in controlling the amplitude of movement.," *Journal of Experimental Psychology,* vol. Vol 47(6), pp. 381-391, 1954.

[11] W. Soukoreff and S. I. MacKenzie, "Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI," *International journal of human-computer studies,* vol. 61, no. 6, pp. 751-789, 2004.

[12] D. Miniotas, O. Spakov and G. Evreinov, "Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space," *IOS Press, (c) IFIP,,* pp. 137-143, 2003.

[13] P. Isokoski, "Text Input Methods for Eye Trackers Using Off-Screen targets," *Proceedings of the Eye Tracking Research & Applications Symposium, New York,* pp. 15-21, 2000.

[14] M. Donegan, L. Oosthuizen, R. Bates, H. Istance, E. Holmqvist, M. Lundalv, M. Buchholz and I. Signorile, "D3.3 Report of user trials and usability studies. Communication by Gaze Interaction (COGAIN)," *Retrieved on-line December, 06, 2015 from: http://www.cogain.org/results/reports/COGAIN-D3.3.pdf,* 2006.

[15] R. Cloud, M. Betke and J. Gips, "Experiments with a camera-based human-computer interface system," in *Proceedings of the 7th ERCIM Workshop "User Interfaces for All", pp. 103-110*, Paris, France, 2002.

[16] R. Hoyt, "StaggeredSpeech," [Online]. Available: http://www.staggeredspeech.org/. [Accessed 24 03 2016].

[17] D. J. Ward, A. F. Blackwell and D. J. C. Mackay, "Dasher - a data entry interface using continuous gestures and language models," in *UIST '00 Proceedings of the 13th annual ACM symposium on User interface software and technology, pp. 129-137*, New York, USA, 2000.

[18] G. C. De Silva, M. J. Lyons, S. Kawato and N. Tetsutani, "Human Factors Evaluation of a Vision-Based Facial Gesture Interface," in *IEEE Computer Vision and Pattern Recognition Workshop, pp. 52*, New York, USA, 2003.

[19] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics- Doklady,* vol. 10, no. 8, pp. 707-10, 1966.

[20] S. MacKenzie, "A Note on Phrase Sets for Evaluating Text Entry Techniques," 03 September 2002. [Online]. Available: http://www.yorku.ca/mack/RN-PhraseSet.html. [Accessed 06 May 2016].

Appendix 1

EVALUATION FORM: A COMPARATIVE STUDY OF TWO LAYOUTS OF A VIRTUAL KEYBOARD FOR HANDS-FREE TEXT TYPING WITH COMPUTER VISION-BASED HEAD POINTING

Rate the techniques using the evaluation scales which are *general evaluation, pleasantness, dominance, quickness, accuracy, efficiency, tiredness, distractibility and mental effort*. The rating scales are so called bipolar rating scales meaning that each end of the scale represents opposite ends of a two way continuum. The scales have nine points varying from -4 to +4. The rating scales vary from negative (left negative side) to positive (right positive side) evaluation and, for example, the more positive the value the more good your experience was. The middle area of each continuum represents a neutral point in between the opposites.

There exists no right or wrong ratings; instead we are interested in knowing your personal evaluations of each keyboard layout used.

1. General evaluation

-4	-3	-2	-1	0	1	2	3	4
	Bad			Neutral			Good	

2. Pleasantness

-4	-3	-2	-1	0	1	2	3	4
	Unpleasant			Neutral			Pleasant	

3. Dominance

-4	-3	-2	-1	0	1	2	3	4

Technique being in control   Neutral   You being in control

4. Quickness

-4	-3	-2	-1	0	1	2	3	4

Slow   Neutral   Quick

5. Accuracy

-4	-3	-2	-1	0	1	2	3	4

Inaccurate   Neutral   Accurate

6. Efficiency

-4	-3	-2	-1	0	1	2	3	4

Inefficient   Neutral   Efficient

7. Physical effort / tiredness

-4	-3	-2	-1	0	1	2	3	4
	Difficult			Neutral			Easy	

8. Distractibility

-4	-3	-2	-1	0	1	2	3	4
	Difficult			Neutral			Easy	

9. Mental effort

-4	-3	-2	-1	0	1	2	3	4
	Difficult			Neutral			Easy	

Appendix 2

FINAL EVALUATION FORM

1. General evaluation

Which keyboard layout was better in general for text typing?

Static	[ ]
Dynamic	[ ]
There were no perceived differences between the layouts	[ ]

2. Pleasantness

Which keyboard layout was more pleasant for text typing?

Static	[ ]
Dynamic	[ ]
There were no perceived differences between the layouts	[ ]

3. Dominance

Which keyboard layout was more pleasant for text typing?

Static	[ ]
Dynamic	[ ]
There were no perceived differences between the layouts	[ ]

4. Quickness

Which keyboard layout was faster for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

5. Accuracy

Which keyboard layout was more accurate for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

6. Efficiency (evaluates time and effort needed to achieve the task)

Which keyboard layout was more efficient for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

7. Physical effort / tiredness

Which keyboard layout was easier for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

8. Distractibility

Which keyboard layout was more distracting for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

9. Mental effort

Which keyboard layout was more mentally difficult for text typing?

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

10. Personal preference

Static   [ ]

Dynamic   [ ]

There were no perceived differences between the layouts   [ ]

11. Is there something you would like to add about your experiences with gaze-based and head-based techniques? Please, write in free form your thoughts below.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Thank you very much!

Dedication and Acknowledgements

I dedicate this thesis to my late father who has been an inspiration for me throughout my life, to my children to whom I want to become an inspiration and to my mother and my wife who have always been there to support and encourage me.

I would like to thank my thesis supervisor Dr. Julia Kuosmanen who continuously guided me during this work. This work could not be accomplished without her great support. I am also grateful to Dr. Oleg Spakov and Professor Markku Turunen for reviewing this work and providing their valuable feedback to improve it.