LiquidKeyboard: An Ergonomic, Adaptive QWERTY Keyboard for Touchscreens and Surfaces

Christian Sax
School of computing and communications
University of Technology Sydney
UTS
Sydney, Australia
sax@it.uts.edu.au

Hannes Lau
Electrical Engineering and Computer Sciences
Technische Universität Berlin
TU Berlin
Berlin, Germany
hlau@cs.tu-berlin.de

Elaine Lawrence
School of computing and communications
University of Technology Sydney
UTS
Sydney, Australia
elaine@it.uts.edu.au

Abstract—**Virtual touchscreen keyboards provide poor text input performance in comparison to physical keyboards, a fact, which can partly be attributed to the weaker tactile feedback, they offer. Users are unable to feel the keys on which their fingers are resting, and usually hover their hands over the keyboard, hitting each key individually. Consequently, users cannot use all ten fingers for typing, which decreases their input speed and causes hand fatigue. We present a keyboard prototype called LiquidKeyboard, which adapts to the user's natural finger positions on a touchscreen and discuss options to allow for the fingers to rest on the screen while typing. When invoked, the keyboard appears directly under the user's fingertips and is able to follow finger movements. As the positions of the surrounding keys are fixed in relation to each finger, users can find and touch the keys without tactile feedback. Additionally the user controlled positioning of the keys allows the keyboard to adapt to the physical specification of the user's hand, such as the size of each finger.**

Keywords-Keyboard; touchscreen; touch surface; adaptive; touch-type; QWERTY.

I. INTRODUCTION

The way in which users interact with computers and electronic devices has changed in recent years as both users and electronic devices become more mobile and wireless connectivity becomes the norm. Ubiquitous computer systems are now a part of everyday life in business, education and social settings. Driven by the increased need for user friendly input interaction devices and powered by the availability of new hardware technology, the input methods used on these devices have deviated from standard-sized physical, tangible keyboards to methods more suitable for the specific physical limitations of mobile devices. Popular and very successful touch-screen phones such as the Apple's iPhone [3] or the iPad tablet computer [4] set the benchmarks in this development, creating a massive shift from physical key or stylus based text input methods to virtual keys (called softkeys) displayed on a touchscreen.

Touchscreen systems employ displays with smooth surfaces that are capable of sensing finger touches, which can then be related to interface elements displayed on the screen directly underneath them. This technology enables direct user interaction with the display, without the need for additional physical keys. For example, a button can be activated when the user touches an image of it on the screen. It can therefore be placed on the interface surfaces without the need to increase the actual device size. As a result, product designers can reduce the number of physical keys to the bare minimum (such as an 'on' and 'off' switch) while, at the same time, increase the screen real estate. Users interact with the interfaces more directly by pushing buttons on the screen, moving objects around, and performing gestures to trigger functions.

On touchscreens and touch surfaces users cannot feel controls even though they can manipulate them by touch. Although the same visual language, such as a button, is used on the user interface one cannot feel the button or its edges. Mobile user interfaces try to compensate the lack of tactile feedback by relying heavily on visual and auditory cues. There are some approaches to make these screen interactions more tangible, such as a screen that changes the surface structure, but these are not yet available as mainstream consumer devices [1, 17, 21].

A. Existing touchscreen text input methods

Textual input with human computer interfaces is measured in two scales: speed and accuracy [15]. Table 1 compares the text input speed in WPM (words per minute) of selected text input systems. These systems can be divided into two groups: firstly, the selective method where users have to select each letter individually and secondly, the predictive method that aim to predict the intended word by analyzing every user input [9]. The mechanical QWERTY keyboard, a selective method system and the most common English keyboard layout, serves as a baseline to compare the text input speeds. WPM figures show the average speed, however, they can vary greatly depending on the user and serve as a guide only.

In comparison to physical keyboards the average input rate on touchscreen keyboards is low (see Table 1). A typical word rate, even with the use of input prediction, is around 15-30 words/min according to [13]. This mainly derives from the fact that devices that are fitted with touchscreens are generally smaller and do not provide sufficient screen space to allow the user to use both hands for typing. But even keyboards on larger devices such the iPad cannot be

 ISBN: 978-1-61208-003-1

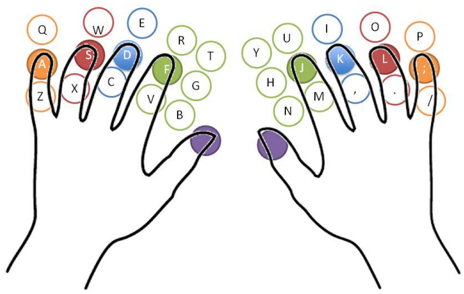

Figure 1 – Showing the keys under the finger contacts

used easily with 10 fingers, as users cannot feel if a finger is on a particular key or not.

On an English QWERTY keyboard layout the fingers are placed on the A-S-D-F and J-K-L-; keys for the left and right hand fingers respectively – these keys are called home keys. Both thumbs rest on the space key. Proficient touch-type writers know where other keys are in relation to the home keys and do not need to look at the keyboard while typing. Yet, on a touchscreen, the tactile guide to keep the fingers on their respective home keys is missing. Without tactile feedback users cannot relate to the key's location and the spatial position of surrounding keys. As a consequence they have to rely on visual orientation, look at the keys to hit the right ones and cannot keep their gaze on the actual task. This increases the eye movements (called saccades), meaning that users cannot perform any action or notice changes on the interface [11] – a lower text input speed and a higher error rate result.

In summary, current touchscreen devices, such as the iPad, do not provide tactile feedback resulting in users needing visual cues to compensate for this lack of tactile feedback. This results in poor typing speeds of around 15-30 WPM. We introduce an alternative to the current available touchscreen keyboards, with which we aim to leverage the ten finger touch-typing and compensate the missing tactile feedback with our adaptive keyboard approach.

After a description of the LiquidKeyboard's concept in Section 2, this paper shows two prototypical implementations, which employ different algorithmic approaches to project the keys under the users fingertips. Section 4 discusses problems that arise when the LiquidKeyboard is implemented on devices that are unable to sense the pressure of a user's touch and presents different approaches to overcome them. Finally, Section 5 discusses plans for future research and implementations followed by the conclusion in Section 6.

II. THE LIQUID KEYBOARD

The design of our new keyboard system, called *LiquidKeyboard*, aims to empower users to utilize all ten fingers on the touchscreen as on a normal physical keyboard. The home keys are displayed under each finger and follow the individual finger position on the screen. Users are free to place their fingers anywhere and do not have to adapt to the straight key rows found on most keyboards. Adjacent keys to the home keys are situated next to them and form groups around fingers (see Figure 1 and Table 2). For instance, the right hand middle finger is on the 'K' home key which forms a group with the 'I' and ',' keys being above and below as on a physical QWERTY keyboard. Groups follow the finger touches on the surface keeping the distance between the home key and the adjacent keys constant. Consequently, the surrounding keys will always be at the same position in relation to the users' finger, no matter where they are on the surface. Users will not have worry about hitting the wrong keys when they shift their fingers.

TABLE I. COMPARISON OF DIFFERENT TEXT INPUT METHODS

Input method	PM	Advantages	Disadvantages	WPM	Applications
Mechanical. QWERTY keyboard	No	-Fast due to physical keys and tactile feedback -All 10 fingers can be used for text input	-Too many keys to be fully supported on mobile systems -Requires physical keys for tactile feedback	64.8 [19]	Desktop computers, phones
Virtual pseudo QWERT keyboard	Yes	-Keyboard layout similar to mechanical QWERTY	-Most of the time very small hence not all keys are supported	18.5 [14]	Tablet computers, handhelds, touchscreen phones
Gesture on pseudo QWERTY virtual keyboard	Yes	-Keys do not need to be hint individually	-User have to learn input method	25 [12]	Touch phones, tablet computers, handheld
Un-constrained handwriting word recognition	No	-Natural handwriting is used as the input method-No vocabulary has to be learned	-Recognition rate can be poor -A stylus is required for most systems	24.1 [11]	Handheld, phones, tablet computers
Single handwritten character recognition	No	-Natural handwriting is used as the input method	-Only one character at a time can be recognised. -Some system require special alphabet	21 [6]	Handheld, Phones, tablet computers

The LiquidKeyboard is designed for touch sensitive surface systems where the screen is big enough to accommodate two hands, such as the iPad. On these systems users can write with both hands while using the 10 finger touch-typing.

Microsoft has patented an input concept where they split a QWERTY keyboard in two halves [16]; one for the left and one for the right hand using the touch of the users' palms on the screen as additional orientation cues. The LiquidKeyboard moves beyond the Microsoft concept having three major advantages. Firstly, the keys follow the individual finger touches movements, secondly palm touches are not required as keyboard orientation reference points (see also Prototype 2 with rotation of key groups) and lastly the keyboard layout adapts to each individual finger/hand shape.

Another LiquidKeyboard benefit is that the interface is easy to learn due to its consistent keyboard layout. As the keyboard can be invoked at any place on the surface, the keyboard adapts to the users' hand physiology such as the hand size and finger position. In forthcoming user tests that compare the LiquidKeyboard to traditional softkey keyboards, we expect to see a decrease in hand fatigue as users can rest their fingers on the screen while typing and do not need to hold them in a hovering position.

We believe our system could improve the usability of mobile emergency system such as the 'Portable Medical Monitoring Computer' [20]. The LiquidKeyboards adaptive capabilities can provide a low cost and effective text input system for different types of touchscreen and touch surface systems.

III. Implementation of the Prototypes

The LiquidKeyboard was implemented in two prototypes to allow empirical testing and to prepare for future project phases. To be able to test on differing platforms we used web technologies, namely HTML, CSS, and JavaScript to create web applications that can run in every Gecko or WebKit based web browser. WebKit specific JavaScript API extensions allow us to harness the multi-touch capability of Apple's iPhone and iPad and react to the touch of multiple fingers. In this first phase of the project both prototypes support only the right hand-side of a QWERTY keyboard, i.e. the home keys are 'J-K-L-;'. The following sections will explain the implementation of the softkey activation and the two different prototypes.

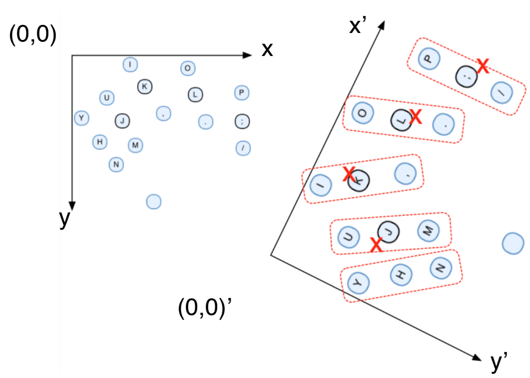

Figure 2 – Original (left) and adapted (right, primed) keyboard layout of the first prototype. The users' initial touches are marked with crosses. The dashed lines depict the key groups that will be moved in unison.

TABLE II. Home keys and their grouping. Bold letters indicate the home keys

Home key	Group
A	Q **A** Z
S	W **S** X
D	E **D** C
F	R T **F** G V B
J	Y U H **J** N M
K	I **K** ,
L	O **L** .
;	P **;** /
SPACE	**SPACE**

A. Key activation

In both prototypes, the keys are defined as points on the plane without spatial extent. After initialization, each touch on the surface is associated to the closest key by a simple nearest neighbour search algorithm [11]. As long as the user's finger remains on the screen the association is maintained and the key is considered pressed. As a consequence users do not have to hit the keys exactly to activate them.

B. Prototype 1 with a keyboard layout transformation

As soon as four touches are sensed on the screen we use these touch positions to place the entire keyboard. To do so, the first prototype has a basic keyboard layout stored in the application, which specifies a position for each key including the home keys. To find an adapted keyboard layout we map the points where the user has touched the screen to the stored positions of the home keys. Then we determine a rotation angle, scale factor and translation vector for a two dimensional Helmert transformation of the stored layout that would bring the home keys from their original positions as close as possible to the positions the user touched (see our current prototype implementations). As the equations to determine the transformation parameters are over determined by the four reference points we use a least-square adjustment as shown in [7] and remain with a rest deviation between the desired coordinates for the home keys and their real positions as images of the transformation applied to the stored layout. The correct association of the four chosen positions to the appropriate home keys is initially unknown, which leads us to consider all possible mappings and chose the one with the lowest remaining deviation.

In a second step, we move each key group identified in Table 2 until the respective home key fits exactly under the user's touch, effectively clearing any deviation from the previous step. This second step, the translation of a key group is also performed each time the user moves a finger resting on a home key. The first step, however, is only used to determine the initial position of the keyboard, once the user touches the screen.

 ISBN: 978-1-61208-003-1

C. Prototype 2 with rotation of key groups

Experiments with the first prototype showed that the keys do not only have to follow the individual finger movements but that the key groups will also need to adapt their orientation based on the hand's position. As the calculation of the transformation parameters proved to be too slow to be carried out every time the user moved a home key we implemented a second prototype with a simpler geometric model, which allows for rotation of the key groups while the user is moving his or her hand on the display.

In this second model, once five touches have been registered (including the thumb), we approximate the user's palm position by calculating a circle with an outline that comes as close as possible to all five touch points in a least-square sense [23]. This allows us to approximate the position and orientation of the user's hand and map the touch positions to the home keys. Ordered on a clockwise circle, the first key after the biggest angular gap is associated with the user's thumb and therefore with the space key while the second touch is mapped to the index finger and 'J' key. All other home keys follow in a clockwise order.

The best-fit circle is used to determine the mapping of the user's fingers the home keys and discarded thereafter. As depicted in Figure 3 we use the apex of an isosceles triangle based on the index and little finger positions to estimate the position of the user's wrist. The angles and side ratios of the triangle are constants and calculated a priori based on the average length (finger tips to wrist = d_1) and breadth (index to little finger = d_2) of the human hand (see (1)) [22].

This method proved to approximate the position of the user's wrist closely enough to align the keys of each group on a ray originating at the wrist position and passing through the group's home key (see also Figure 3). With a measured distance d_1 and the constant c the second LiquidKeyboard prototype is able to update the wrist position and rotate the key groups fast enough to parallel the user's hand movements.

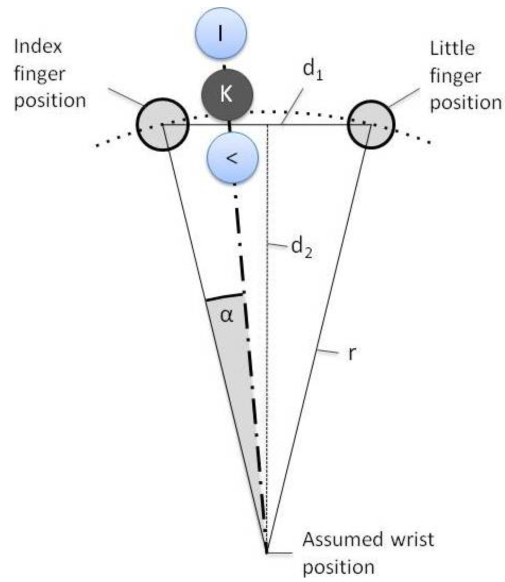

Figure 3 – Shows the rotation of the K home key group with a given index and little finger position

IV. Discussion

A. The keyboard layout

Experiments with 4 different users have shown that both prototypes have a good layout adaption of home keys to the sensed finger touch positions on the screen. The key activation is done by the nearest neighbour search algorithm rather than by sensing touch events within a defining geometric area (such as a rectangle or circle) representing a key. This solution helps with keyboard layouts where keys were spread out because keys are still activated if the sensed touch is close enough to the key but would be out of range of the visual indicated key.

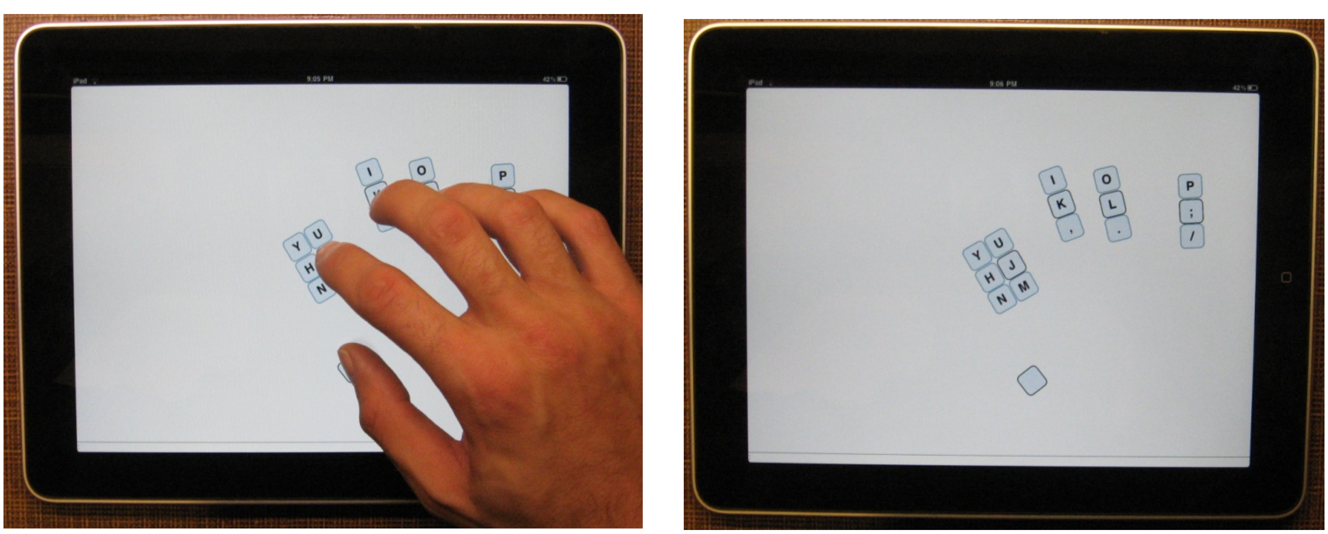

Figure 4 – The left photo shows the second LiquidKeyboard prototype on an Apple iPad. The right picture shows how the keyboard adapts to the shape of the user's hand.

 ISBN: 978-1-61208-003-1

$$c = \frac{d_1}{d_2} \quad (1)$$

$$c \approx 0.47 \quad (2)$$

The findings of the first prototype, which adapted only to the fingers' positions but not to different wrist positions, were taken forward into the next prototype generation. By adding an algorithm that rotates the key groups according to sensed index and little finger position a more user friendly and ergonomic keyboard layout was reached. The assumptions made on the wrist position in the described equation and geometric setup (see (2)) rotate the key groups in an adequate angle that match the actual human anatomy in our trials. Having the adjacent keys pointing towards the user's wrist resulted in an ergonomic type feeling with keys being at the right spot.

The detection of the index and little finger touch necessary for the rotation and the realignment computation of the key groups had no negative impact on the keyboard's responsiveness.

B. Activation of the home keys

Experiments with both prototypes have shown that the activation of the home keys is problematic. Fingers are resting on the display, which allows the algorithm to sense the touch positions and adapt the keyboard layout accordingly. This poses two usability challenges for the activation of the home keys: First, while other keys will only be touched with intent to activate them, the home keys will also sense touches of fingers returning to the home position after activating a key in the same key group. These touches however are not meant to activate the home keys. Secondly, it is desirable for the user to be able to activate any home key while his or her finger is resting on it. On a physical keyboard this is possible by changing the finger pressure and depressing the key. However, most current touchscreen systems are unable to determine the finger pressure of a touch.

In a naïve implementation where any touch triggers activation and any activation requires a touch, users would have to lift a finger and place it back in order to activate a home key on which they previously rested their finger. Thus the input recognized by the LiquidKeyboard would be incomplete; instead of the intended word 'kilogram' the input 'iogrm' would be read. Furthermore, if users return their finger to a home key after activating one of the adjacent keys, the home key would unintentionally be activated as well, adding additional letters to the recognized word. With both effects considered a naïve implementation would read the word 'kilogram' as 'i**k**o**l**gr**f**m**j**' (additional letters in bold).

Obviously the output of such naïve implementation would be far from the desired result. In the following sections we propose and discuss multiple solutions for the LiquidKeyboard, which overcome these usability challenges and are close to the well-known input paradigm of a physical keyboard hardware solution.

With touchscreens that are capable of sensing pressure users could increase their finger pressure in order to activate a home key on which their finger was already resting. In the keyboard would sense that the pressure is not high enough to activate the key. There are pressure sensitive touchscreen patents by Apple [2] and technologies by Peratech [18], which have the potential to solve this problem with a hardware solution. However, for touchscreen devices without the capability to measure the finger pressure employed by the user touching the screen, software solutions can be applied.

1) Sensing pressure with increasing touch surface area.

In our current prototype, implementations a sensed touch is recognized as two coordinates describing the position of a single point on the surface. The information passed to the keyboard is therefore independent from the actual touch area on the touchscreen, i.e. no matter how big or small the finger is the result will be a single point. The touch area of a finger on a screen increases when the finger is pressed harder against the surface. Hence the resting finger and a finger that is actively pressing against the screen is different; the latter will have larger touch area. We intend to leverage this effect for our home keys to sense whether users are resting their fingers on the screen or are activating one of them.

2) Alternative keyboard layout solution.

On devices that are unable to sense the pressure or covered area of a users' touch the keyboard layout could be modified. By moving the home keys in front of the users' fingertips the user would be able to activate them just like any other key by moving his or her fingers to the keys position and touching it. After doing so the user could return his finger to the previous home position without unintentionally activating a key there.

3) Dictionary solution.

Alternatively predictive text algorithms can be used to associate the input string recognized by the keyboard. For instance the input "ikolgrfmj" would be mapped to the English word "kilogram". If the mapping is ambiguous and multiple words exist whose input would be recognized as "ikolgrfmj" the input context could be used to determine the word the user intended to type.

We believe that the best user experience will be achieved by sensing the touch pressure, as this more closely resembles the user interaction paradigm closest to a normal mechanical keyboard. However, until the required hardware becomes available, sensing the increase of the finger touch area with increased pressure or the use of a specialized predictive text algorithm seem to be the best solutions with the current technology at hand.

V. FUTURE IMPLEMENTATION

The keyboard should not only adapt to the users' finger and hand positions but also to frequently miss-hit keys. In our current prototypes each touch to the surface is associated with the closest key, thus the user does not need to hit the displayed keys exactly. In order to make the keyboard more adaptive, the key found to be closest to the users touch can be moved partway towards the touch location, making future attempts to find the key at the same location more likely to succeed. With this mechanism the key layout will adapt to the users' writing style as in a similar implementation by [8].

We will extend the current prototypes to provide a complete QWERTY layout and support input for both hands as a basis for forthcoming user testing. This will allow us to compare the performance of the LiquidKeyboard to existing input methods with regard to their input speed and accuracy. Our third prototype will measure the effect of the solutions that we presented for the home key touch differentiation problem on the overall user experience and to identify further possible usability issues related to the proposed interaction methods.

VI. CONCLUSION

The LiquidKeyboard leverages a widespread and commonly used text input concept (the QWERTY keyboard layout) and makes it usable on touchscreens and touch surface interfaces. Our approach compensates for the lack of tactile feedback on smooth touch sensitive surfaces by making the keyboard adapt to the finger position and by following finger touch points. Users can rest their finger on the screen as on a mechanical keyboard and do not have to worry about the keys positions as they follow the user's fingers.

In the discussion we identified the home key activation problem as a challenge for our touchscreen keyboard and addressed it with solutions in soft- and hardware. Investigating these ideas and testing and comparing them in greater detail will be a field for further research.

REFERENCES

[1] Apple Computer Inc., *Keystroke tactility arrangement on a smooth touch surface*, United States Patent application number: 20070247429, Accessed on 24/12/2010 from http://www.faqs.org/patents/app/20090315830

[2] Apple Computer Inc., *Force Imaging Input Device and System*, Patent number: 20070229464, Accessed on 24/12/2010 from http://www.patentstorm.us/applications/20070229464/claims.html

[3] Apple Computer Inc., *iPhone*, 2010, Accessed on 10/6/2010 from http://www.apple.com/iphone/

[4] Apple Computer Inc., *iPad*, 2010, Accessed 10/6/2010 from http://www.apple.com/ipad/

[5] E. Clarkson, J. Clawson, K. Lyons, and T. Starner, An empirical study of typing rates on mini-QWERTY keyboards. In *Extended Abstracts on Human Factors in Computing Systems* (CHI '05, Portland, OR, USA, April 02-07 2005), ACM, New York, NY, pp.1288-1291, Accessed on 24/12/2010 from doi: 10.1145/1056808.1056898

[6] M. D. Fleetwood, M. D. Byrne, P. Centgraf, K. Q. Dudziak, B. Lin, and D. Mogilev, An Evaluation of Text-Entry in Palm OS - Graffiti and the Virtual Keyboard. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting 2002*, pp. 597-601.

[7] C. D. Ghilani. *Adjustment Computations: Spatial Data Analysis*, Wiley, 5th Edition, March 2010, Accessed on 24/12/2010 from doi: 10.1002/9780470586266

[8] J. Himberg, J. Häkkilä, P. Kangas, and J. Mäntyjärvi. On-line personalization of a touch screen based keyboard. In *Proceedings of the 8th international Conference on intelligent User interfaces* (IUI '03, Miami, Florida, USA, January 12-15 2003). ACM, New York, NY, pp.77-84, Accessed on 24/12/2010 from doi: 10.1145/604045.604061

[9] P. Isokoski. *Manual text input: experiments, models, and systems*. Doctoral dissertation, University of Tampere, Finland, 2004. Electronic dissertation, Accessed on 7/12/2010 from http://acta.uta.fi/pdf/951-44-5959-8.pdf

[10] R. J. Jacob, Eye tracking in advanced interface design. In *Virtual Environments and Advanced interface Design*, W. Barfield and T. A. Furness, Eds. Oxford University Press, New York, NY, 1995, pp.258-288.

[11] D. E. Knuth, *The art of computer programming*, Vol. 1 (3rd ed.): fundamental algorithms, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, 1997.

[12] P. Kristensson and L. C. Denby. Text entry performance of state of the art unconstrained handwriting recognition: a longitudinal user study. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (CHI '09, Boston, MA, USA, April 04-09 2009). ACM, New York, NY, pp.567-570, Accessed on 24/12/2010 from doi: 10.1145/1518701.1518788

[13] O Kristensson, Five Challenges for Intelligent Text Entry Methods, *AI Magazine*, Vol. 30, No. 4, 2009. Accessed on 10/6/2010 from http://www.aaai.org/ojs/index.php/aimagazine/article/view/2269/0

[14] M. H. Lopez, and I. S. MacKenzie, and S. Castelluci, *Text Entry with the Apple iPhone and the Nintendo Wii*, 2009. Accessed on 10/9/2010 from www.malchevic.com/papers/iphone_paper.pdf

[15] I. S. MacKenzie and R. W. Soukreff, Text Entry for Mobile Computing- Models and Methods, Theory and Practice. In *Human-Computer Interaction*, Vol. 17, Issue 2&3 September 2002, pp.147-198.

[16] Microsoft Corporation, *Virtual keyboard based activation and dismissal*, Patent number US20090237361, Filed 24/9/2009, Accessed on 10/6/2010 from http://www.patents.com/virtual-keyboard-based-activation-and-dismissal-20090237361.html

[17] Nokia, *Tactile Touch Screen*, Patent number PCT/ET2006/009377, Filed 27/9/2006, Accessed on 10/6/2010 from http://www.unwiredview.com/wp-content/uploads/2008/07/nokia-tactiles.pdf

[18] Peratech Ltd., *Paratech homepage*, 2010, Accessed on 10/6/2010 from http://www.peratech.com/index.php

[19] H. Roeber, J. Bacus, and C. Tomasi, Typing in Thin Air: The Canesta Projection Keyboard - a new Method of Interaction with Electronic Devices. In *CHI '03 Extended Abstracts*, pp. 712-713.

[20] C. Sax and E. Lawrence, Tangible Information: Gestures for a Portable e-Nursing touch screen interface. In *Procceedings of the Healthcom 2009 11th International Conference* (16-18 December 2009), pp.1-8, Accessed on 24/12/2010 from doi: 10.1109/HEALTH.2009.5406211

[21] Staska, *Nokia haptikos tactile touchscreen details emerge,* 2008, Accessed on 10/6/2010 from http://www.unwiredview.com/2008/07/08/nokia-haptikos-tactile-touchscreen-details-emerge/

[22] A. R. Tilley and Henry Dreyfuss Associates, *The Measure of Man and Woman: Human Factors in Design*, John Wiley & Sons, 1993.

[23] D. Umbach and K. N. Jones. A few methods for fitting circles to data. *IEEE Transactions on Instrumentation and Measurement, Vol. 52, Issue* 6, 2003. pp.1881-1885.