University of Strathclyde

Department of Computer and Information Sciences

Novel Text Entry and Mobile Interaction Techniques for Arabic Language Users

By

Karim Mohsen Mahmoud El Batran

A thesis presented in fulfillment of the requirements for the degree of
Doctor of Philosophy

2015

Dedication

Dedicated to

My Father

Who has taught me that dedication and hard work are the essential for a successful life

My Mother and Sisters

Who always inspire me with their love, support, and prays

My Wife

Who always encouraged and inspired me

My Big Family

Who always were there all along my lonely trip

Karim El Batran

Declaration

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination that has led to the award of a degree. I have referenced the work of others where appropriate throughout the thesis.

Acknowledgements

First, I would like to express my greatest gratitude to the Almighty ALLAH, the only one who has given me the opportunity, strength, courage and blessing to pursue this work. However, conducting the research and writing this thesis required the patience, persistence and motivation of many people whom I would like to personally acknowledge. First, I would like to extend my deepest thanks to my supervisor, Dr. Mark Dunlop, for his patience, help and guidance in completing this research provided me endless source of thoughts and inspiration. His enthusiasm in research, systematic organisation in work and positive attitude towards life confidently influenced my study and life. I am also grateful to Prof. Dr. John Sorghan and Dr. Eva Hornecker for their guidance and direction. Second, I am thankful to the 145 participants who took part in the studies reported here, without their time and input the research would not have seen the light.

Third, many other people- too many to mention individually- contributed in one-way or another to this study. To all of you, please accept my sincere and profound gratitude. Last, but certainly not least, these acknowledgements would be incomplete without expressing my deepest appreciation to my family. Special acknowledgement of this achievement goes to both my father and mother, who not only showed me unconditional love, acceptance, and respected the person that I am throughout my life, they also showed me the strength of character which has brought me to where I am today. My sincere thanks are also to my sisters, future wife, big family, parent-in-laws, and brother-in-laws for their constant support and prayers. I am so lucky and so proud to have such a wonderful family. Without them, my Ph.D. could have been an extremely lonely trip.

Abstract

Inspired by an observational study of Egyptian Agricultural Census counters, this research aims to improve mobile data entry though better form navigation and improved Arabic text entry. Four improvements were taken into consideration in sequence: (1) minimizing large forms to fit small mobile device screens and easing form navigation process, (2) optimizing Arabic keyboard layout to suit Arabic Language users, (3) introducing Gesture-based Arabic Writing Pads (GBAWPs) that fit small mobile device screens and smart watch surfaces, and (4) enhancing a quantitative prediction model to overcome the defect in modeling interactions on mobile devices.

This research shows an improvement of form navigation on mobile devices. The approach is based on computerizing forms and using Panning and Zooming as a navigation technique. In order to do so, an observational study was conducted on the Egyptian Agricultural Census (EAC). However, there were considerable challenges in reducing the size of the paper forms to fit mobile devices and introducing fast navigation technique. It was concluded after computerizing the forms that using the Panning and Zooming technique scored less completion task time and workload in comparison to the tabbed navigation technique.

Moreover, this research presents a new design of an Arabic keyboard layout for effective text entry on touch screen mobile phones. The approach is based on Pareto front optimization using three metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for Arabic Language users through approximate alphabetic sorting. In user studies, the new layout showed an observed improvement in typing speed in comparison to a common Arabic layout. Currently, there is an opportunity to research new optimized keyboard designs with less usage experience than QWERTY as in mainstream Western European languages. Pareto optimization can produce high quality keyboards for alphabet based languages that could be beneficial when there is less reluctance to change from QWERTY.

Furthermore, this research also illustrates Gesture-based text entry as a method used for mobile devices. Its success and acceptance is critically dependent on the reliability of gesture recognition. The gesture recognition of the GBAWP is accomplished through a sequence of touched points or swipes on the screen. In order to maximize the text area field and minimize the number of keys displayed on the screen, a 12-key GBAWP interface was introduced appearing like a 12-key physical keypad phone. Considering the Arabic letters characteristics, structure, and maximizing speed, a 6-key GBAWP layout based on dot recognition was introduced. After conducting usability tests on both the 12-key and 6-key GBAWP, it was found that users could perform text entry on mobile devices using the 12-key GBAWP with an estimate of 2.9 words-per-minute on average. They also executed text entry tasks on a Sony SmartWatch 2 with an average of 3.2 words-per-minute. This could increase to an estimate of 4.5 words-per-minute on average, on the long term. While entry speeds were slow, users found it easy to use and it supports largely eyes free interaction. Gesture-based technique enables users to perform Arabic text entry on small display mobile devices and watches using both the 12-key and 6-key GBAWP.

Finally, this research introduces an enhancement to KLM (Keystroke-Level Model), a quantitative prediction model predicting the user's behaviour in low-level tasks. This was acomplished by extending it with three new operators describing interactions on mobile touchscreen devices and tablets. The approach is based on Fitts' Law to identify a performance measure estimate equation for each of the introduced interactions. Three prototypes were developed to serve as a test environment in validating Fitts equations and estimating the parameters for these interactions. Three-thousand and ninty observations took place with a total of 51 users. The studies confirmed that most interactions fitted well with Fitts' Law. On the other hand, it was noticed that Fitts' Law does not fit well on small mobile device screens when the Index of Difficulty exceeds 4 bits. These results enable developers of mobile device and tablet applications to describe tasks as a sequence of operators used and predict user interaction times prior to creating prototypes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

HCI	Human–computer Interaction
UCD	User-Centered Design
EAC	Egyptian Agricultural Census
KLM	Keystroke-Level Model
ICT4D	Information and Technology for Development
GBAWP	Gesture-Based Arabic Writing Pads
PDA	Personal Digital Assistant
T9	Text on 9 keys
ID	Index of Difficulty
MT	mean Movement Time

1. Introduction

1.1 Motivation

Human–computer Interaction (HCI) involves the study, planning, and design of the interaction between users and computers. This interaction arises from introducing new techniques that help the user to accomplish the task in an easy, fast, and accurate way. Understanding the mentality, ability and need of users is the core of developing and improving these techniques. Accordingly, the user was placed at the center of the design and the concept of 'User-Centered Design' (UCD) was introduced.

UCD is a broad term to describe design processes in which end-users influence the design outline. It is a broad philosophy with a variety of methods. There is a spectrum of ways in which users are involved in UCD, but the important concept is that the users are involvement in a one way or another (Abras, et al., 2004; Black, 2008). A user-centered approach is particularly useful when a new technique is introduced or when a step-change in an existing technique is required.

This research investigates improving form navigation and form fill-in behavior for Counters of the Egyptian Agricultural Census (EAC), introducing optimized keyboard layouts for Arabic users, providing new Gesture-based text entry technique on small display mobile devices for Arabic language, and enhanceing KLM (Keystroke-Level Model), a quantitative prediction model predicting, by extending it with three new operators describing interactions on mobile touchscreen devices and tablets.

An observational study was conducted on the Egyptian Agricultural Census (EAC). It targeted the techniques used in running the census process, such as; forms fill-in, data entry, and the process's accuracy. Throughout the observations and discussions with senior counters, it was found that most counters could be categorized as having low to zero levels of experience dealing with computerized devices such as; PCs, laptops, tablets, and/or smart-phones. For reference, the percentage of computer users was approximately 23% in Egypt in July 2010. This percentage increased to 42.78% in July 2013 (Ministry of communication and Information Technology, 2013).

Although, there is technological progress in information-gathering systems and data collection, the Egyptian Ministry of Agriculture and Land Reclamation still collects the data manually on large (double sided 100x35 cm) paper forms in books (40 forms per book). The Ministry then transfers the books to the headquarters, and then enters the data to a database system using "Data Entry" employees – separate from the counters – primarily because of the low technical background of the counters. This double entry has been identified as affecting the accuracy of data and increasing the cost of the census.

However, regarding text entry, mobile devices are equipped with virtual keyboards having no physical buttons. Letters are selected by tapping the screen of a mobile device. Past researchers have proposed many techniques to optimize keyboard layouts for mobile devices. However, the suggested techniques were massively dominated by English language for text entry, keyboard layout, and statistical models, while there was a lack in research targeting the optimization of keyboard layout for other alphabet-based languages such as Arabic.

Additionally, Gesture-based text entry is a method used for mobile devices. In this method, gestures are drawn instead of tapping keys on the screen. The gesture recognition for the gesture-based text entry is accomplished through a sequence of touched points or swipes on the screen. Yet, a minimum number of researches focused on introducing text entry on small screen devices such as watch surface. Moreover, researches targeting the introduction of gesture-based text entry for other languages such as Arabic were never tackled.

Finally, in order to evaluate a new technique or an existing one, a quantitative prediction model must be introduced. KLM is the model that could be used to estimate the time taken to complete simple data input tasks. Past researches have proposed many enhancements for KLM to evaluate different techniques. Still, KLM cannot totally describe new interactions used on mobile devices and tablets nowadays.

1.2 Problem Statement

While, there is technological progress in information-gathering systems and data collection, yet the Egyptian Ministry of Agriculture and Land Reclamation still collects the data manually. Introducing more computerized data collection using electronic forms and improved form navigation on mobile devices led to interesting questions such as; what is the attitude of the counters towards computerizing the form? Will counters face challenges upon computerizing the form? How will the change of size from large forms to tablets affect interaction? Will the need to navigate through the form affect external working memory of the counters? And do Pan and Zoom techniques better support their working memory?

On the other hand, there is a technological progress in text entry, especially in optimizing keyboard layout, still there is an absence in research targeting the optimization of keyboard layout for other alphabet-based languages such as Arabic. There are questions that need to be answered as; what is the attitude and ability of the Arabic Language users for text entry? How will they deal with the optimized Arabic keyboard layout? Is the optimized Arabic keyboard more familiar regarding letters position than the Apple Arabic keyboard layout? And will it support fast and accurate text entry?

Even though the Gesture-based text entry method was introduced, there is still the problem of researches not targeting languages such as Arabic, or not considering the characteristics of the character as a target in the optimization process. This also led to the introduction of some questions such as; will this method support fast and accurate text entry? What are the expected entry speeds? How far can this method for Arabic text entry be compared to alternative English text entry methods? And will this method enable Arabic text entry on small screen mobile devices?

Additionally, as mentioned previously there are few enhancements on quantitative prediction model, such as KLM, which enable it partially to estimate the time taken to achieve certain new interactions done on mobile devices and tablets. This takes the research to answer some important questions such as; is KLM capable of handling more interactions done on mobile devices and tablets? Do the new operators describing these new interactions fit into Fitts' Law? Did previous research introduce precise prediction equations used to estimate the time taken for certain interactions? And can KLM be the model used to evaluate newly developed keyboard layouts, despite of the language, for mobile devices and tablets?

1.3 Research Objective

The research aims at implementing improvements, which enable the users to carry out the data entry process efficiently using mobile devices and tablets. This is done first by observing the user, who is a counter of the EAC using Arabic Language. Afterwards, introducing a computerized data collection using electronic forms with improved form control and navigation on mobile devices. This comes in the phase of improving Form fill-in area, as shown in figure 1.1.

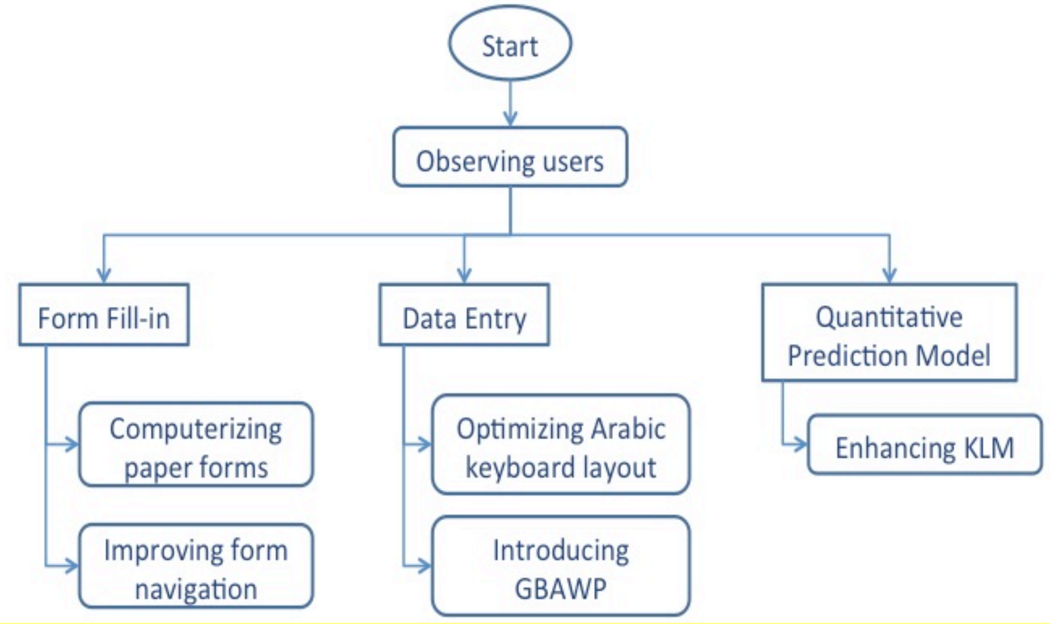

Start
Observing users
Form Fill-in
Data Entry
Quantitative Prediction Model
Computerizing paper forms
Improving form navigation
Optimizing Arabic keyboard layout
Introducing GBAWP
Enhancing KLM

Fig 1.1 Flow of the research

Furthermore, increasing data entry rate by optimizing the Arabic keyboard layout located on mobile devices. The approach is based on Pareto front optimization using three metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for Arabic Language users through approximate alphabetic sorting. Moreover, by developing a 12-key Gesture Based Arabic Writing Pad (GBAWP) interface appearing like a 12-key physical keypad phone, in order to maximize the text area field and minimize the number of keys displayed on a watch screen. In addition, by presenting a 6-key GBAWP layout based on dot recognition, which takes into consideration the Arabic letters characteristics, structure, and maximizing speed.

Finally, after presenting an enhancement of form navigation using Panning and Zooming on mobile devices, an optimized Arabic keyboard layout, and the 12-key and 6-key GBAWP. An enhanced quantitative prediction model must be introduced to evaluate these techniques and keyboards, the research enhances KLM, by extending it with new operators necessary to describe three interactions used on mobile devices and tablets.

1.4 Research Methodology

Based on the nature of this research, it was mainly intended to investigate and improve text entry and mobile interaction techniques for Arabic language users. Inspired by observations of the Egyptian Agricultural Census process, the first part enhances the form navigation using Panning and Zooming on mobile devices. The second part is all about improving the text entry field through developing a new optimized Arabic keyboard layout. When it comes to the third part, it constitutes a presentation of a 12-key and a 6-key GBAWP. Eventually, the last part discuses the improvement of the quantitative prediction model KLM to prepare it to be able to handle tasks on mobile devices and tablets. Qualitative methods and user studies, design and iterative prototyping, and evaluation, all known in HCI research over the years (Dix et al., 2003), were used respectively to accomplish the aim of this research. Overall, the research abided by phases of qualitative analysis to understand users, followed by the development of mobile prototype system. Eventually, this leaded to formal usability experiments of the prototypes.

Although there is technological progress in information-gathering systems and data collection, the Egyptian Ministry of Agriculture and Land Reclamation still collects the data manually on large paper forms. The Ministry then transfers the books to the headquarters, and then enters the data to a database system using “Data Entry” employees. This double entry has been identified as affecting the accuracy of data and increasing the cost of the census. From here the idea of introducing a computerized data collection method, using electronic forms with improved form control and navigation on mobile devices, was initiated.

A qualitative research was done in the form of a field observational study of the EAC for data collection. The aim of the study was to escort 10 EAC counters, while collecting field data through personal interviews with the holders in their places of residence and filling several items on paper forms. Handwritten notes regarding the process flow and photos were taken. Accordingly, two electronic prototypes were created based on those items using two techniques; selecting and typing. As for the selection technique, the first prototype used Panning and Zooming navigation while the other used Tabbed navigation. Regarding the typing technique, the standard Arabic keyboard layout was used for the text entry process. The electronic versions of prototypes were implemented using HTML5 and JavaScript. The standard Safari browser was used to display the electronic prototype versions using an Apple iPad 2.

A number of EAC counters, nominated by and worked at the Egyptian Ministry of Agriculture, were asked to go through the evaluation process by executing preplanned tasks on both prototypes. Their age, gender, study, and work varied, while all were bachelor degree holders. The dependent variables were task-completion time (in seconds), interface preference, and subjective workload. The results were statistically analyzed using T-test. Moreover, questioning users through semi-structured interviews and NASA Task Load Index questionnaire were used to extract user's preferences.

Accordingly, it was noticed that there is difficulty in handling text entry tasks using the available Arabic keyboard layout. However, this lead to the need for improving the text entry process by developing a new optimized Arabic keyboard layout. Dunlop and Levine (2012) introduced an optimization technique for English keyboard layouts based on Pareto front optimization. In fact, the Pareto front optimization technique was used in this research to optimize keyboard layouts of languages rather than English, such as Arabic, in order to improve text entry speed.

A qualitative research was done in the form of an observational study for data collection. Volunteering native Arabic speakers were asked through semi-structured interviews to give their preferences/recommendations regarding two suggested paper keyboard layouts. Subsequently, prototype developing took place, upon users preferences, using Pareto front optimization technique to develop the layout for the Arabic keyboard. The electronic version of prototypes was implemented using Java displayed using safari on HTC Desire S.

Evaluation was carried out with the help of volunteering native Arabic speakers who were asked to go through a set of preplanned procedures. Although their age, gender, study, and work varied, still they were all bachelor degree holders living and working in Egypt. The dependent variable was words-per-minute. Results were analyzed statistically using T-test. In addition, questioning users through semi-structured interviews was done to extract user's preferences.

Based on the above-mentioned observations, the idea of decreasing the input area for non-English text entry to fit small devices using gesture based technique was introduced. Wobbrock, Myers, and Kembel (2003) presented EdgeWrite technique that provides stable text entry for people with motor impairments. Moreover, Dunlop (2004) introduced an English text entry system for the use on a watch face. By merging both techniques and understanding the Arabic letters characteristics and structure, a new GBAWP was introduced in this research to provide text entry with high speed.

A qualitative research study was done for data collection. Volunteering native Arabic speakers, living and studying in the United Kingdom, were asked to perform a study on text entry fragmentation for Arabic language, using pen and paper, to record text entry speeds. Accordingly, a 12-key GBAWP interface appearing like a 12-key physical keypad phone was developed. This came in the context of maximizing the text area field and minimizing the number of keys displayed on the screen. Moreover, considering the Arabic letters characteristics and structure together with maximizing speed, an optimization technique based on dot recognition was used to create a 6-key GBAWP. This pad is designed to fit on smaller display mobile devices such as watch screens. The electronic versions of prototypes were implemented using HTML5, Java, and JavaScript displayed using safari on Apple iPad 2, an Apple iPad Mini, and a SONY SmartWatch 2 connected to a Motorola Moto G via Bluetooth.

Consequently, after developing the 12-key and 6-key GBAP prototypes, volunteering native Arabic speakers were asked to carry out a set of preplanned tasks for the evaluation process. The age, gender, study, and work of participants varied, while all were bachelor degree holders living and working in Egypt. The dependent variable was words-per-minute. Results were analyzed statistically using T-test and evaluated using enhanced KLM-GOMS model.

After presenting an enhancement of form navigation using Panning and Zooming on mobile devices, an optimized Arabic keyboard layout, and the 12-key and 6-key GBAWP, a quantitative prediction model must be introduced to evaluate these techniques and keyboards. Models, such as GOMS (Goals, Operators, Methods, and Selection rules) and KLM have been shown to be useful tools in modeling interaction and in deciding between, and filtering out designs (Card et. al, 1983).

The development of three prototypes came in the process of enhancing KLM derived from GOMS model. These prototypes reflected three main interactions used nowadays on tablets and mobile devices (Swipe, Tap, and Zoom). They were built to estimate the time taken to conduct these interactions. Fitts' Law (Paul Fitts, 1954) was used to analyze the general case of estimated time taken for these interactions. The electronic versions of prototypes were implemented using HTML5 and JavaScript displayed using safari on an Apple iPad Mini.

Volunteering participants were asked to go through the evaluation process by executing preplanned tasks. Their age, gender, study, and work varied, while all of them were bachelor degree holders living and working in Egypt. Additionally, three equations were formed to estimate the movement time for three interactions; Swipe, Tap, and Zoom.

In summary the techniques used were:

Chapter 3: Improving form control and navigation on mobile devices

- In-field observation of an EAC census officer.
- Development of two iPad prototypes using HTML5 and JavaScript.
- Questionnaire and controlled usability test with 20 EAC census officers.
- Statistical analysis of timing and workload data.

Chapter 4: Optimizing Arabic Text Entry on Mobile Devices

- An observation of 10 Arabic Language users using two different Arabic keyboard layouts.
- Semi-structured interviews to extract users preferences.
- Development of an android prototype using Java.
- Questionnaire and controlled usability test with 40 users.
- Statistical analysis of timing data.

Chapter 5: Gesture-Based Arabic Writing Pad

- An observation of 15 Arabic Language users on text entry fragmentation for Arabic language.
- Semi-structured interviews to extract users preferences.
- Development of prototypes for iPad and SONY Watch using HTML5, Java, and JavaScript.
- Questionnaire and controlled usability test with 30 users.
- Statistical analysis of timing data.

Chapter 6: Enhancing KLM on Mobile Devices

- Development of three iPad prototypes using HTML5 and JavaScript.
- Controlled usability test with 30 users.
- Statistical analysis of timing data.
- Formation of three equations to estimate the movement time for three interactions using Fitts Law.

1.5 Report Organization

The dissertation is divided into seven chapters. Chapter one is an introduction defining the main points of the research. Chapter two, a literature review will be introduced to give an overview of the importance of the techniques utilized in the research. It will cover UCD methodology, text entry concept on mobile devices, and KLM-GOMS model for mobile device interactions. Chapter three, a study was done regarding the improvement of form control and navigation on mobile devices. Chapter four illustrates an informal survey on key distribution of Arabic keyboard, and the way of optimizing the Arabic keyboard layout. Chapter five, a Gesture-based text entry technique on small display mobile devices will be presented, were Arabic text entry was performed on small display mobile devices and Sony SmartWatch 2 using both the 12-key and 6-key GBAWP, respectively. Chapter six, an enhancement on KLM model to handle interactions on mobile devices and tablets will be introduced, were 51 users participated in conducting three user studies with a total of 3090 observations. Chapter seven, a final conclusion is drawn with a future work plan together with key contributions. Finally, references and appendices are presented.

2. Literature Review

Human–Computer Interaction (HCI) involves the studying, planning, and designing the interaction between users and computers. Indeed, S. Card, T. Moran, and A. Newell were the first researchers to define this term in their book, "The Psychology of Human-Computer Interaction" in February 1983. The HCI term indicates that a sort of an open-ended dialog arises between the user and the computer (Card, et al., 1983). Interaction between users and computers occurs at the user interface, where objects can be displayed on the screen, input received via keyboards and mouses from user, or other user interactions with large-scale computerized systems such as aircraft and power plants (Myers, 1998).

Because Human–Computer Interaction studies a human and a machine in conjunction, it relies on supporting knowledge from both. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human factors such as computer user satisfaction are important (Grudin, 2007).

One of HCI basic goals is to make computers more usable and receptive to the user's needs. Furthermore, systems are designed to minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task (Sears and Jacko, 2007; Jacko and Sears, 2003). Therefore, researchers in HCI are interested in developing new design methodologies for interfaces, introducing techniques for evaluating and comparing interfaces, experimenting with new hardware devices, prototyping new software systems, exploring new paradigms for interaction, and developing descriptive and predictive models and theories of interaction (Norman, 1988). Their work often revolves around not only designing graphical user interfaces but also web interfaces.

The study, design, and development of interaction for mobile devices and services have become an area of significant research field in the past years, in which mobile devices and technologies have only been available for less than a decade. These devices and technologies are largely spread without any training, leading the focus of HCI researchers to contribute in such a young field.

Since the emerging of Mobile HCI field, research has focused mainly on the improvement of techniques and devices themselves: how to accommodate small screens, how to make devices smarter, how to design faster input mechanisms, how to establish more reliable communications, etc. Nowadays, a new approach is taken into account, which adds the larger social and contextual factors surrounding mobile device to the study, design, and development process.

This chapter gives a quick review on literature. It is divided into three sections. In the first section, ICT4D in developing countries is illustrated. In the second section, brief description of text entry concepts are given. In the final section, GOMS model and specifically KLM (Keystroke-Level Model), a quantitative prediction model, are introduced.

2.1 ICT4D in developing countries (case study in Egypt)

Information and Technology for Development (ICT4D) utilises technologies, an example of which are personal computer, mobile devices, and the Internet, to improve the socio-economic status of developing countries. The "Development" term in ICT4D not only refers to elimination of economic poverty from third world developing countries, but also works on the improvment of the quality of life for poor societies in rich countries (Toyama, 2010).

Users in ICT4D are different from those who are known by HCI researchers. In 2011, total world population was estimated at 6.9 billion. Most of HCI researches targeted the 1.6 billion PCs users (Computer Industry Almanac, 2012) and neglected the rest. Despite the fact that this latter population are unfamiliar with PCs, they are becoming rapidly familiar with other computing mobile devices such as mobile phones. A total of around 6 billion active mobile-phone accounts in the world, mostly in developing countries, was sighted between 2010 and 2011 (ITU, 2012). Accordingly, this has helped with the introduction of Mobile HCI research in ICT4D.

In Egypt, as a study case, it was found that more than half the Egyptian population could be categorized as users of minimal technological background. That is to say, they are perceived as having low to zero levels of experience dealing with computerized devices such as PCs, laptops, tablets, and/or smart-phones. For reference, the percentage of computer users was approximately 23% in Egypt in July 2010. This percentage increased to 42.78% in July 2013 (MCIT, 2013). As mentioned previously, a user-centered approach is useful when introducing new technique. Therefore, considering their mentality, ability, and need was the inspiration behind seeking enhancement in interaction handled on mobile devices, and it is important to ground the research in the intended context of use.

This section introduced Information and Technology for Development (ICT4D), focusing more on users in developing countries such as Egypt. The next section will include an overall introduction on text entry concepts, the history of optimizing text entry on mobile devices for English language, history of Arabic language, a quick review on research done for the Arabic language text entry, and a glance on text entry for small display mobile devices.

2.2 Text Entry Concept

Typing is the process of inputting text into a device, such as; typewriter, mobile phone, and computer, by pressing keys on a keyboard. Typewriters had been invented as early as 1714 by Henry Mill and reinvented in various forms throughout the 1800s. Christopher Sholes was the inventor of the world's first practical typewriter. In 1874, Sholes & Glidden typewriters established the “QWERTY” layout for the letter keys.

The QWERTY layout was introduced not to reduce typing speed but to minimize the risk of mechanical jamming of the keys. The jam was cause due to the hit of two neighboring keys at the same time. The reordering of keys came by distributing frequent letter pair combinations to both hand sides of the keyboard, which decreased the probability that two keys next to each other would be hit at the same time and consequently increased the efficiency of typing speed. Although various layouts of keyboards were introduced, still the high entry rate, the reasonable low error rate, and the society’s trend sticking to good technologies made “QWERTY” reins standard for English-language keyboard layout for typewriters, mobile devices, laptops and desktops. (David, 1985; Gould, 1992).

2.2.1 Text Entry on Mobile Devices

Due to a long time-trace history, consumer’s interest in small mobile devices has driven the need for more efficient, portable, and text input methods. Due to the need of minimizing the size of the device, the difficulty of allocating more space for text entry aroused. This challenge was a push to produce new keyboard designs that require smaller and more efficient physical layouts and that can be optimized for a one or two finger typing instead of two-handed typing.

Past researchers have proposed many techniques to enable users to perform text entry as quickly and precise as possible. They also examined key positioning and size. Sears, et al. (1993) found that smaller keys reduced text entry speed and increased errors. MacKenzie et al. (1995) presented an evaluation of different keyboards. Still, it was found that, although a smaller keyboard increased errors compared to a larger one, there was no reduction in speed. MacKenzie et al. (1999 and 2001) presented a model which predicts the maximum (expert) and minimum (novice) typing speeds for a given keyboard layout with stylus typing.

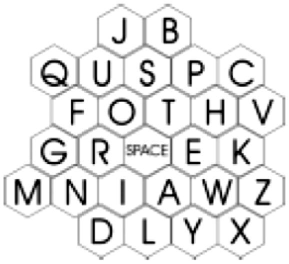


Fig 2.1 Metropolis I keyboard layout (Hunter et al., 2000)

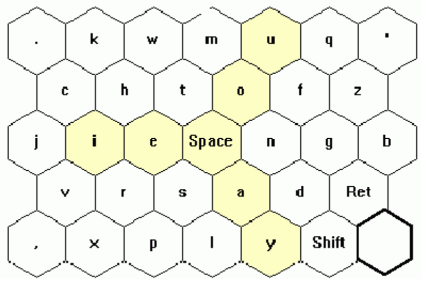
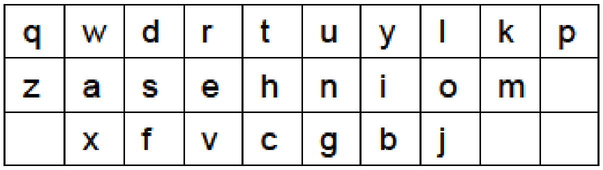


Fig 2.2 Metropolis II keyboard layout (Zhai et al., 2000)

Zhai, Hunter and Smith (Hunter et al., 2000; Zhai et al., 2000) took a physics-based approach to find an optimized keyboard arrangement, which was done by applying the Metropolis algorithm. Hunter et al. (2000) presented a Metropolis keyboard, as shown in figure 2.1. Moreover, Zhai et al. (2000) presented Metropolis II, another Metropolis-derived keyboard as shown in figure 2.2, in a later publication. In further work, Zhai et al. (2002) compared their Metropolis keyboard, to other common layouts, and presented a user evaluation.



q	w	d	r	t	u	y	l	k	p
z	a	s	e	h	n	i	o	m	
	x	f	v	c	g	b	j		

Fig 2.3 Bi, Smith and Zhai's Quasi-Qwerty layout (Bi et al., 2010)

Bi et al. (2010) introduced an alternative layout to overcome the QWERTY layout. They reordered the QWERTY layout by shuffling keys by at most one position from their original location to achieve a quasi-optimized QWERTY variant. As shown in figure 2.3, its layout was quite near to the QWERTY layout, which gave it a privilege since user would expect the keys position.

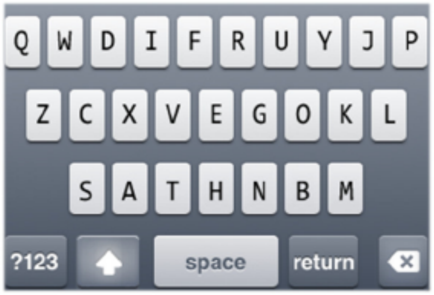


Fig 2.4 The Sath-Trapezoidal keyboard (Dunlop and Levine, 2012)

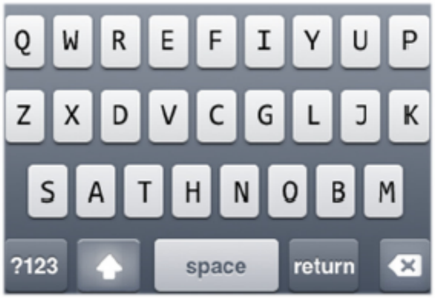


Fig 2.5 The Sath-Rectangular keyboard (Dunlop and Levine, 2012)

Dunlop and Levine (2012) introduced a new optimization technique for keyboard layouts based on Pareto front optimization. This Optimization attempts to optimize for speed of text entry, and familiarity to the traditional Qwerty layout. It has also been extended to reduce the likelihood of hitting a neighboring key that will result in a valid dictionary word, and thus improve error correction and accuracy. They presented the concept of badgrams – pairs of letters that when one is substituted for another often result in valid words (e.g. *I* and *O* on a QWERTY keyboard leads to words such as *in/on*, *sin/son*, and *fir/for* being differentiated by very small keyboard distances). A small improvement in speed with a considerable improvement in the tap interpretation metric was found when compared both keyboards, shown in figure 2.4 and 2.5, with the quasi-optimized keyboard. Table 2.1 compares the metric scores for the standard and the rectangular layouts.

Table 2.1 Comparing standard and the rectangular layouts (Dunlop and Levine, 2012).

	Finger Distance	Neighbor Ambiguity	Qwerty Familiarity	Average Score
Sath-Trapezoidal	0.694	0.695	0.694	0.694
Sath-Rectangular	0.751	0.751	0.751	0.751

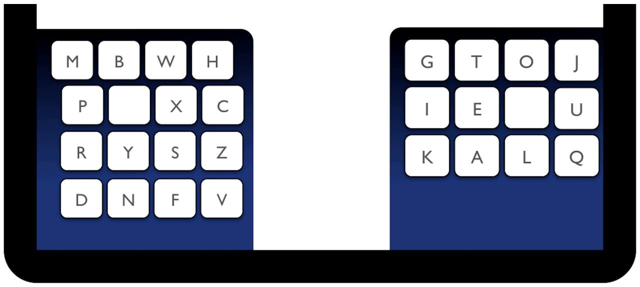


Fig 2.6 KALQ layout (Oulasvirta et al., 2013)

Oulasvirta et al. (2013) studied the design of split keyboards for fast text entry with two thumbs on mobile touchscreen devices. KALQ design was based on landscape-oriented device usage. Letters were divided almost equal, at 54% and 46% for the right and left thumb, respectively. The right thumb handles all vowels, except “y”, while the left thumb has most of the consonants, as shown in figure 2.6. The spacebar is placed on either side as a blank key so as to favor quick switches and minimize travel distance. They reported that KALQ is superior to the quasi-QWERTY layout by 4.1% and to the alphabetical layout by 6.1%. Studies showed that KALQ layout increases typing speed by 34%. This increase in the number of words-per-minute was reached after, on average, 16.8 hours of training.

Recently, most of this work has focused on text entry on mobile devices. Unfortunately, experimental data on text entry performance reported in the literature varies widely due to the use of different performance metrics and experimental designs. Hence, it is difficult to compare studies or to extract meaningful average text entry speeds and error rates. This makes it hard for designers and researchers to use and apply these results and work against the synthesis of a larger picture. Moreover, all the suggested techniques focused mainly on the English language in text entry, keyboard layout, and statistical models.

2.2.2 Gesture-based Text Entry Methods for Small Display Mobile Devices

The two dominant text entry methods found on mobile devices are handwriting recognition and virtual keyboard. Handwriting recognition recognizes and changes user’s handwriting characters, either by stylus or finger, to computer format. Virtual keyboard is an image of a keyboard displayed on the mobile device screen, and it recognizes a character when a key button is touched on the image. Indeed, only one or two fingers are used in virtual keyboard. Technology continuously tries to make devices smaller and more compact. Nevertheless, after the enhancement of the hardware technology, the size of mobile devices continuously decreases and that causes accuracy problems. When it comes to virtual keyboard, it is difficult to hit the exact buttons due to the small size of each key button. Moreover, handwriting recognition method is still not yet efficient to recognize various people’s handwritten characters, so it causes recognition accuracy problems. Because, handwriting recognition needs a complex algorithm, therefore it does not fit the low power mobile devices.

Gesture-based text entry was introduced as another method of text entry for mobile devices. In this method, gestures are drawn on the screen instead of touching keys. Handwritten characters can be gestures instead of letters. Gesture-based text entry uses symbols and metaphors such as; edges, points, and spaces. This method could hinder the user's writing on the screen. However, it is the best when it comes to recognition accuracy and speed because it guarantees a correct output when user writes a correct gesture on the screen. Typical gesture-based text entry methods are EdgeWrite, Quikwriting, and Cirrin.

2.2.2.1 Gesture-based Text Entry Techniques

Stylus is a well-known way of interaction with mobile devices. Methods such as gesture recognition (e.g. Graffiti or Jot), which need drawing symbols of individual characters on the screen, offer beneficial ways of entering text. However, there are some disadvantages to these methods. The pitfalls of gesture recognition revolve around certain facts such as;

1. A user has to know which pen strokes symbolize a certain character, instead of interpreting the user's handwriting by the device.
2. Outside of the human interaction component, the device spends considerable processing time in recognizing each character, which hampers the efficiency of text input.
3. A certain dedicated area where characters are drawn, may be needed on the device itself thus claiming screen space that might be used for other purposes.
4. The creation of gestures need an excess amount of accuracy, and due to the shrinking size of the screen, it might become more difficult.

Moreover, methods such as Jot, Graffiti, and many other inspiring designs have been introduced for stylus-based text entry on small devices. An example of a stylus-based gesture recognition technique is EdgeWrite (Wobbrock, Myers, & Kembel, 2003). EdgeWrite provides stable text entry for people with motor impairments. Users enter text by traversing the edges and diagonals of a square overlay placed over the normal text input area. Character recognition is achieved through the sequence of corners that are hit. Examples of EdgeWrite gestures are shown in Figure 2.7. Potential drawbacks include the need to learn different gestures and the room required to draw them.

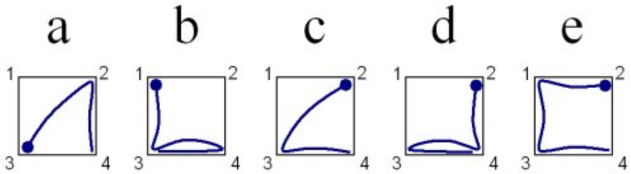


Fig 2.7 EdgeWrite text entry system, gestures for a-e (Wobbrock et al., 1998).

Figure 2.8 shows the Cirrin System (Mankoff & Abowd, 1998). Cirrin is a word-level unistroke text input system. Users form a word by moving their stylus from letter to letter of the word without lifting the stylus from the input surface. Testing showed that Cirrin is as quick as other existing pen entry systems. It still uses 26 cells to represent the English alphabet, which needs a significant amount of screen to implement. Moreover, due to the small relative size of the cells, the task of choosing a letter gets harder and requires precise operation of the stylus to enter a word. As screen size shrinks, the movement between letters becomes more difficult causing more errors.

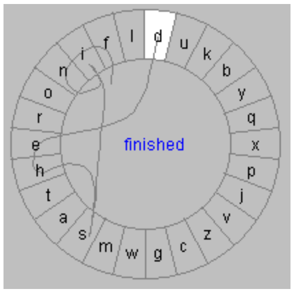


Fig 2.8 Using the Cirrin text entry system to enter the word "finished" (Mankoff & Abowd, 1998).

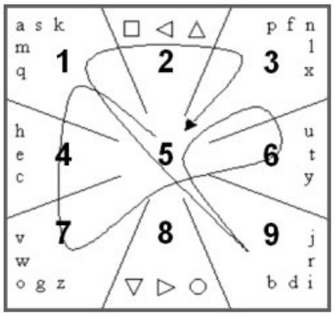


Fig 2.9 Using the Quikwriting text entry system (Perlin, 1998).

Perlin developed a system called Quikwriting (1998). The system has nine virtual keys with multiple characters on a single key. A stylus is used to slide through the desired keys, similar to the Cirrin system. Every time the stylus slides across a virtual key, the key is recorded. The final key sequence is then matched to a pre-stored dictionary of words, as shown in figure 2.9. Abiding by this method, the user does not stop moving the stylus pen. A 3x3 grid is used in the system, and each grid means a letter zone.

ShapeWriter allows users to gesture the shapes of words by connected lines that sequentially intersect the center points of the letter keys of the word over an on-screen keyboard, as shown in figure 2.10. Each word in a large lexicon has a predefined template gesture. A pattern recognizer is used to identify the user's intended word by looking at the scale-translation invariant shape similarity between the user's gesture and the set of predefined template gestures (Zhai and Kristensson, 2003; Kristensson and Zhai, 2004; Kristensson, 2007).

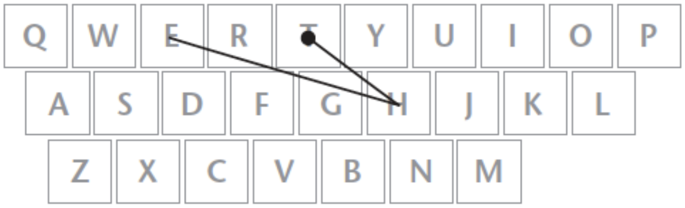


Fig 2.10 The Ideal Shape of the Word "the" in ShapeWriter (Kristensson, 2007).

Seunghun and Geehyuk (2008) introduced a gesture-based Korean text entry method as an alternative text entry technique to provide high accuracy and intuitiveness on handheld computers with small screens such as PDA.

2.2.2.2 Small Screens based Text Entry Techniques

MacKenzie (2002) presented six techniques for three-key text entry. Characters are arranged in six different sets and found by rotating a wheel through the chosen set. Left and right arrow keys are used to maneuver a cursor over a linear sequence of letters and a select key enters a letter. Each of the six techniques had a different set of arranged characters and/or a different cursor mode. Character sets were arranged: alphabetically with SPACE at the beginning of the set, alphabetically with SPACE at the middle of the set, and dynamic rearrangement of letters after each entry using a linguistic knowledge. The cursor mode was either persistent (It maintains its position after each character entered) or snap-to-home (whereby the cursor jumps to the SPACE character after each entry).

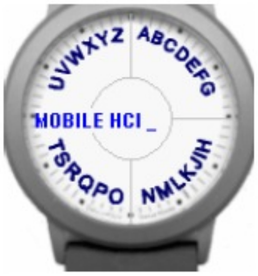


Fig 2.11 5-button watch top text entry system (Dunlop, 2004).

The text entry system shown in figure 2.11 was made for the use on a watch face. It has four alphabetic buttons and a central space button (Dunlop, 2004). This system works as a dictionary based disambiguation text entry system, similar to that of T9 (Text on 9 keys, a patented predictive text technology for mobile phones). For each character of an intended word, the key that contains the character is pressed once. The sequence of key presses is then compared to those stored in a dictionary. The space key is used to cycle through all possibilities to compose the intended word. User studies have shown encouraging results that demonstrate the feasibility of such a system in spite of the fact that the 5 buttons occupy much of the watch face.

Haggerty and Tarasewich (2005) presented a stylus-based text entry method named Touch-Point. This method does not require additional screen or the use of complex gestures. The method features a variable-speed scrolling character set operated by the stylus. Characters are invoked by placing the stylus anywhere on a display, thereby allowing text entry in a space as small as a single character.

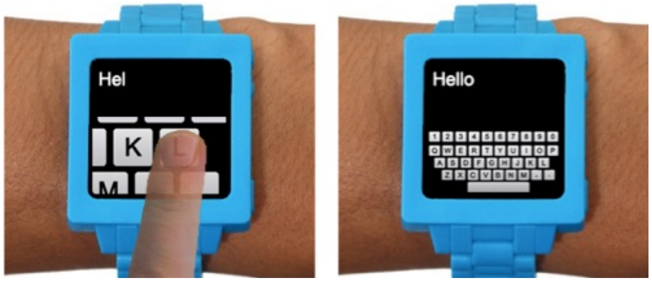


Fig 2.12 ZoomBoard on a watch-sized device (Oney et al., 2013).

Oney et al. (2013) present a soft keyboard interaction technique called ZoomBoard that enables text entry on ultra-small devices as shown in figure 2.12. They used zooming technique to enlarge QWERTY layout tiny keys displayed on screen to comfortable size in response to user presses.

Participants achieved, on average, 7.6 wpm on their first use of ZoomBoard, and 9.3 wpm by the final trial. However, participants using the nonzooming keyboard achieved 4.5 wpm.

2.2.3 Arabic Language History and Keyboards

Arabic is one of the oldest living languages in the world. It comes in the fourth position among languages worldwide. It is the main language for 22 countries in North Africa, Western Asia (Middle East), and East Africa. The total number of Arabic language speakers is 452 million (representing 4.23% of world population) from which 293 million are natives (Nationalencyklopedin, 2010 and Arabic reference at Ethnologue, 2013).

2.2.3.1 The Origin of the Arabic Language

The Arabic language is the mother tongue of most of the countries in the Middle East such as; Saudi Arabia, Egypt, Lebanon, Syria, Sudan, Libya, Tunisia, Morocco, Algeria, and many other countries. In his article "A history of the Arabic Language", Brian Bishop says that the Arabic language is one of the most important languages because over 150 million people speak it and also it is indispensable for many Muslims walking on the face of this planet (Bishop, B., 1998). In addition, the Arabic language is the sacred language of Islam due to its usage in Qur'an. Therefore, many Arabs are proud of speaking this language. Another proof of how significant the Arabic language is to the Arabs, is the history of the Arab world that was full of many foreign invasions and occupations. That is to say, during those invasions and occupations, many countries were trying to impose their own languages on the countries they occupied. However, this fact has not affected the importance of the Arabic language and its position amongst the Arabs. Not only did they refuse to embrace any language other than the Arabic, but also, they kept speaking their native language proudly. This section will try to track down the origin of this language and shed the light on its importance in Arabs daily lives.

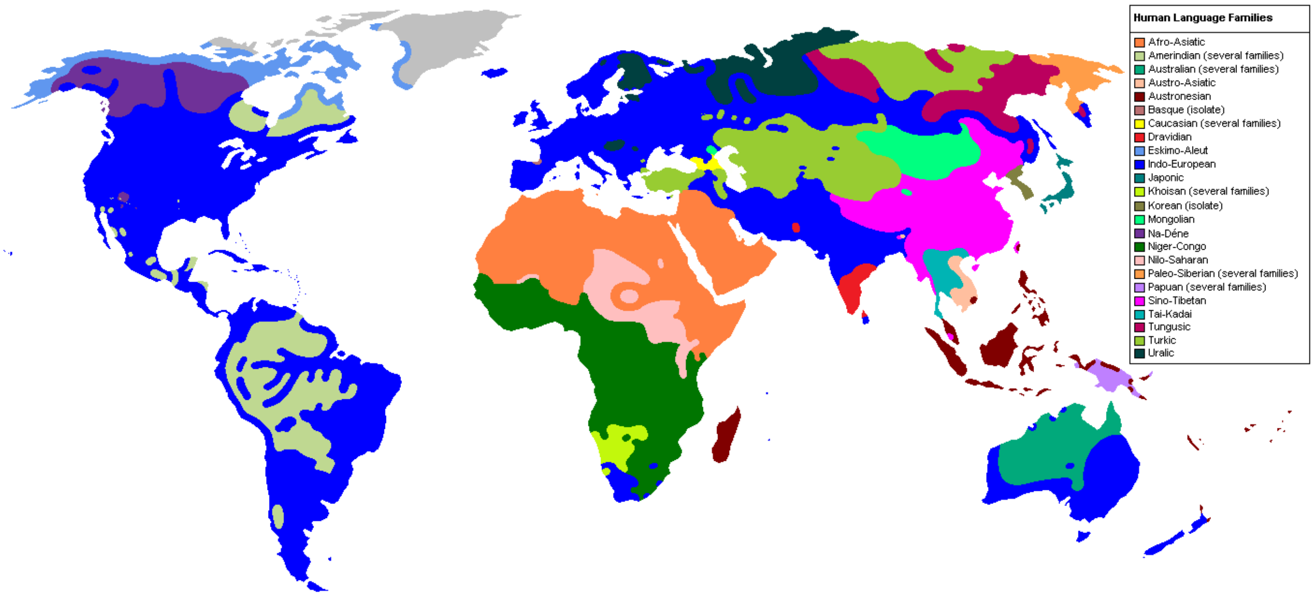


Fig 2.13 Primary Human Language Families Map (Primary Human Language Families Map, 2013).

The "Guide to the World's languages" book written by Merrit Ruhlen explains that Arabic is one of the Semitic languages (Ruhlen, M., 1987). To be specific the author says in his book that "Arabic is a part of the Semitic subgroup of Afro-Asiatic languages". Therefore, defining what does Semitic mean is a must for us to grasp the history of the Arabic language. Tapani Harviainen, a professor of Semitic Languages at the University of Helsinki, elucidates in his article "Semitic Languages and (the Phoenician Language)", that the Semitic language comprises three groups which are; Southern Semitic, Western Semitic, and Eastern Semitic. He adds that the three groups are interrelated but every group belongs to a certain area. For instance, the Eastern Semitic belongs to Mesopotamia, while, the Western Semitic belongs to the Middle East such as; Lebanon and Syria, and finally, the Southern Semitic refers to the Arabian Peninsula and Ethiopia. This means that the Southern Semitic is the origin of the Arabic language because Arabic was mainly spoken at the area of the Arabian Peninsula. Actually, this is true because Professor Tapani Harviainen mentioned in his article that the Southern Semitic is divided into three groups; the South Arabia language, Arabic, and the Ethiopian languages. In addition, Harviainen states that the origin of Arabic lies specifically in the central north of the Arabian Peninsula.

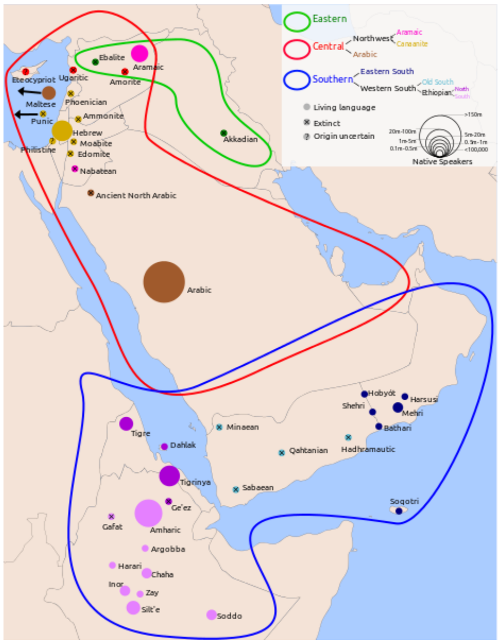


Fig 2.14 Arabic languages within Semitic languages (Semitic languages, 2011).

This previous piece of information makes things more obvious now. That is to say, if we focus on the area of the Arabian Peninsula we will be able to understand how did the Arabic language spread widely that fast. This area was the cradle of the pre-Islamic poetry that was written in Arabic and all the poets back then were known and famous for their meticulous and immaculate choice of Arabic vocabulary while writing their poems. They were also known for their ability of improvisation perfectly and quickly. All that indicates they used to have a grip on the Arabic language. That is why when god has chosen Mohamed, who was from the Arabian Peninsula, to be the prophet of Islam, he gave him the responsibility to teach his nation the Qur'an that was written in the same exact language that the people around him excelled at which is Arabic. In other words, the Qur'an was written in Arabic so as to be understandable by the community that surrounded prophet Mohamed.

When prophet Mohamed and his fellows started to extend Islam throughout many different areas, the Arabic language flourished and started to spread among the different nations because as we said before it was the sacred language of Islam. After being conquered by the Muslims, many different countries gave up on their languages and started learning Arabic instead so as to adapt and conform to the changes that occurred. However, Professor Tapani Harviainen says in his article "Semitic Languages (and the Phoenician Language)", that the adaptation and conformity to the Arabic language imposed the creation of different dialects. But, the presence of all these dialects does not affect the written Arabic language.

After knowing that the Arabic language came from the Southern Semitic language, now a questione must be asked which is, "What is the origin of the Southern Semitic language or maybe the origin of the Semitic language itself?". According to Merrit Ruhlen in his book, "Guide to the World's Languages", it is stated that Arabic is a member of the Semitic subgroup of the Afro-Asiatic group of languages (Ruhlen, M., 1987). In order to understand the previous, an interview that was made with Christopher Ehret, a scholar of African history and African historical linguistics and a professor at the University of California at Los Angeles, should be mentioned (Ehret, C., 2004).

This interview was done by the World History Connected (WHC). Within this interview there is a part that will help us track down the origin of the Southern Semitic language. In this part Ehret is trying to explain how the Semitic language did move from Africa to the Middle East so he said, “The Semitic language group was the result of African migration into Palestine”. Ehret also said that African migrants already speaking proto-semitic moved into the Middle East, introducing and spreading the language. After they spread the proto- semitic it developed into Semitic. This means that the developing of the Semitic language was created in the Middle East. Therefore, now we can tell that the origin of the Arabic language is Africa.

Reaching the conclusion that Africa is the origin of the Arabic language, this was reinforced by an article written by Meredith Holt called," The Afro-Asiatic language family" (Holt M., 1999). According to Meredith Holt stating from "Atlas" and specifically page 51, she says, "Written records of Egyptian and Semitic (both Afro-Asiatic languages) date back at least four thousand years, giving the Afro-Asiatic language family the longest history of any known language group." Meredith Holt adds that the Afro-Asiatic language is mainly divided into five main languages or families as she refers to them and they are; Berber, Chadic, Cushitic, Omotic, and Semitic.

In a nutshell, now we can say that the Afro-Asiatic family language is the mother of all languages. It is so intriguing that a language can develop to be a completely different language. Also, most of the nations succeeded in creating their own dialect that is derived from a certain language. What is strange is that we can find two dialects derived from the exact same language yet the speakers of both dialects could sometimes fail to understand each other. For example, if we look at the language of the Levant and the language of Morocco, they are both derived from the Arabic language but sometimes Moroccans fail to understand what the Lebanese people say. However, when it comes to the written Arabic they are the exact same.

The Arabic writing system flows from right-to-left and is always cursive; both when printed and handwritten. There are 28 basic letters in the Arabic alphabet. However, additional letters and symbols were added in order to accommodate the needs of other languages. Each letter has multiple forms, it is drawn in an isolated form when written alone and is drawn in up to three other forms when connected to other letters in the word. The position of the letter in the word identifies its form. As an example, there are 4 forms for the letter Ain: isolated (ع), initial (عـ), medial (ـعـ), and final (ـع). These four forms have similar frequencies in Arabic text: isolated 23.4%, initial 27.8%, medial 21.0%, and final 27.8% and the average Arabic word has 4.3 letters (Abandah and Khedher, 2009).

2.2.3.2 Arabic Language Keyboards

The common Arabic keyboard layout is derived from the Arabic typewriter keyboard's layout. In the 1970s and 1980s, many computer companies such as; Apple, Sakher, AMEER, ALIS, and Nafitha, developed more than 20 variants for this layout. However, these different variants had consistent allocation for some letters, with differences in allocating other letters particularly the letters in the lower row.

The Arab Standardization and Metrology Organization (ASMO) developed a standard Arabic keyboard layout. This keyboard supports the ASMO standard for the 7-bit Arabic characters code shown in figure 2.15 (ASMO, 1985; ASMO, 1987). However, the market adopted the currently used Arabic keyboard layout shown in figure 2.16 and ignored the standard ASMO keyboard. After adoption of Microsoft to this keyboard layout, it gained wide acceptance in PCs and servers over other layouts for Arabized products. Also other Arabic keyboard layout currently in use, such as Apple MAC, is the one shown in figure 2.17.

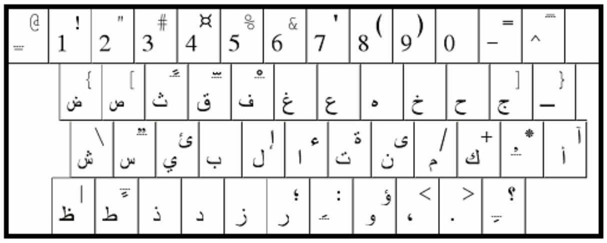

@ ! 1 " 2 # 3 ¤ 4 % 5 & 6 ' 7 (8) 9 0 = - ^
ض { ص [ث ق ف غ ع ه خ ح ج] — }
ش \ س ي ئ ب ل إ ا ء ت ة ن ى م / ك + * أ آ
ظ ط ذ د ز ر ؛ : و ؤ ، < . > ؟

Fig 2.15 ASMO 663 Arabic Keyboard Layout

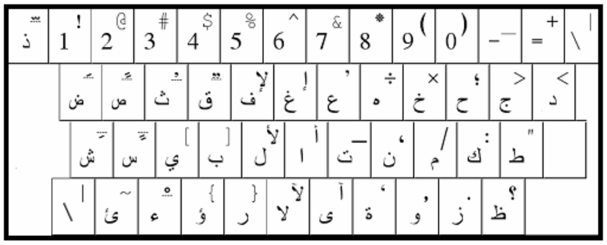


Fig 2.16 Common Arabic Keyboard Layout

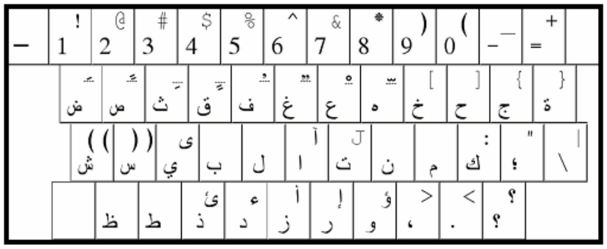


Fig 2.17 Apple MAC Arabic Keyboard Layout

There has been no research on optimizing Arabic text entry on mobile devices. However, on desktops, Idlebi and Mrayati (1990) designed an efficient Arabic keyboard based on statistical approach. Khorshid et al. (2010) used a genetic algorithm to optimize the design of the Arabic keyboard layout for typing speed as shown in figure 2.18. Each keyboard arrangement is evaluated based on the six criteria listed by Yannou and Hossenlopp (2000): load location, number of keystrokes, hand alternation, consecutive usage of the same finger, avoid large steps, and hit direction. The final score of the keyboard is determined as a weighted linear combination of these six individual criteria. Their studies showed a 36.3% speed improvement over the present PC Arabic keyboard. It was also used to evaluate the optimized layout against other layouts including the common layout. Since Apple MAC Arabic Keyboard Layout was the leading one on physical keyboards, it was adopted on touch screen phones preventing the rise of any other optimal keyboards, same as QWERTY layout in English language.

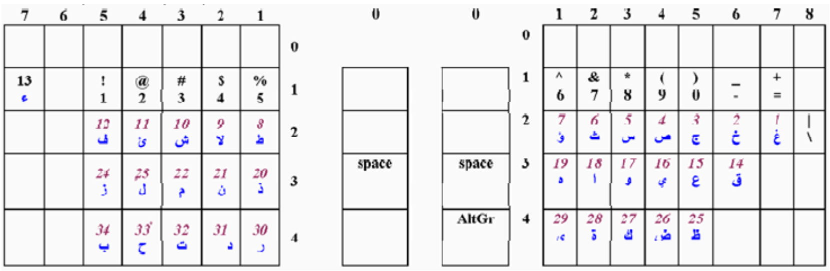


Fig 2.18 The optimized arrangement for Arabic language keyboard (Khorshid et al., 2010)

This section showed an overall introduction of text entry concepts, history of optimizing text entry on mobile devices for English language, a glance on text entry for small display mobile devices, history of Arabic language, and. a quick review on research done for the Arabic language text entry. The next section introduces GOMS model and specifically KLM (Keystroke-Level Model), as a quantitative prediction model. It gives a brief description of GOMS components, structure, and variations. Additionally, KLM's history will be introduced as a mobile quantitative prediction model.

2.3 GOMS Model

GOMS is a specialized human information processor model used to analyze the user complexity of interactive systems. In other words, it attempts to model and predict the user's behavior involved in human-computer tasks. The GOMS model was developed by Card, Moran and Newell, as a way of quantitatively predicting the skilled and error free performance of users interacting with a text editor (Card et. al, 1983). It is used to reduce a user's interaction with a computer to its elementary actions. Moreover, such actions are used to study a certain interface. These actions can be physical, cognitive or perceptual.

2.3.1 GOMS Components

There are several GOMS variations that allow for different aspects of an interface to be accurately studied and predicted. The major concepts remain the same with the presence of those different variations. However, this concept is divided into four components:

- Goals: are what the user intends to accomplish. These can be defined at various levels of abstraction.
- Operators: are elementary perceptual, motor or cognitive actions whose execution is necessary to accomplish the goals. They are described as atomic elements in the model, with specific output and specific duration. That is to say, it is generally assumed that each operator requires a fixed amount of time for the user to execute. This time interval is independent of context.
- Methods: are procedures for accomplishing a goal. In other words, are described as algorithms that the user has internalized in order to determine the sequence of sub-goals and operators necessary to achieve the desired goal. However, they are set to be learned

procedures that the user already has at performance time. Moreover, they are not plans that are created during a task performance.

- Selection rules: are the representations of the control structure in the model used to describe when a user would select a certain method over the others to satisfy a given goal. They generally take the form of a conditional statement. Conversely, these selections should not be problematic and should proceed smoothly and quickly.

2.3.2 GOMS Model Overview

The model owes its ancestry to a production system/goal decomposition approach to cognitive modeling. However, the control structure has been considerably simplified by representing the model as a stack machine. This does not allow it to adequately deal with interruptions and errors, but considerably simplifies it. In addition, because operators are assumed not to be concurrent, only linear processes can be modeled (Card et. al, 1983).

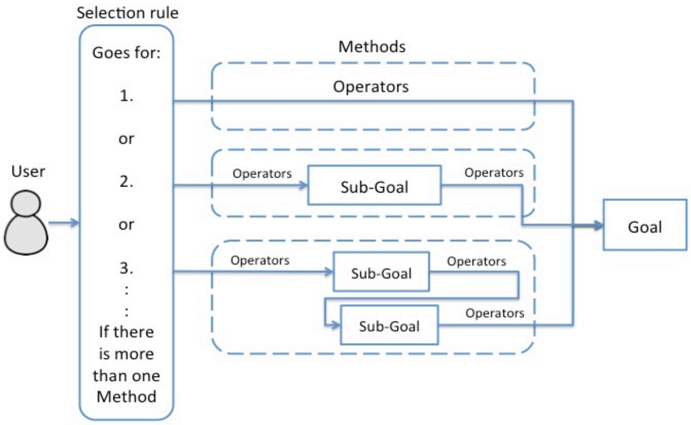


Fig 2.19 GOMS model process flow chart

The user intends to accomplish a certain main goal. As shown in figure 2.19[1], methods are sets containing operators; others contain one or a sequence of sub-goals and operators to accomplish the intended final goal. There may be more than one method available for the user to accomplish the goal. Therefore, the role of selection rules appears in selecting the best method, in terms of how smooth and quick. On the other hand, it may be that task environment features dictate that only one method is appropriate.

[1] The flowchart shown in figure was illustrated by the candidate.

2.3.3 GOMS Model Variations

The plain GOMS first introduced by Card, Moran and Newell is now referred to as CMN-GOMS. Several variations have been proposed to address issues within the original model. Card, Moran and Newell also introduced Keystroke Level Modeling (KLM) as the next GOMS technique (Card et. al, 1983). It made several simplifying assumptions that make it just a restricted version of GOMS. The third major variation on the GOMS technique is the 'Natural GOMS Language' or NGOMSL. This technique gives a very strict language for building GOMS models. The final variation of GOMS is CPM-GOMS. This technique is based on the Model Human Processor. The main advantage of CPM-GOMS is that it allows the modeling of parallel information processing by the user.

2.3.3.1 Card, Moran, and Newell GOMS (CMN-GOMS)

As mentioned previously, CMN-GOMS is the original version of the GOMS technique in human computer interaction. It was developed in 1983 (Card et. al, 1983). It can predict operator sequence as well as execution time. It is represented in a program form, where the tasks are in a pseudo-code format. This makes it amenable to analysis as well as execute. One of its uses is to estimate the load the task places on the user.

The CMN-GOMS method assumes that a user comprehends information in the following manner:

- Eyes/ears perceive information
- Information enters perceptual processor
- Information enters the visual/auditory image store
- Information is stored in the working memory and long term memory
- Information is analyzed in the cognitive processor and a desired reaction (motor function) is chosen
- Desired motor function is activated in the motor processor
- Desired motor function is applied by user's body

All measurements are provided in the following form: middleman [fastman, slowman]. The "middleman" term is the most typical time it takes to complete the action. The "fastman" is a "best case" scenario were time is expected to be the best that a user could possibly do. On the contrary, the "slowman" time is a "worst case" scenario.

In CMN-GOMS, the following Methods-Time Measurement data should be used:

- Eye fixation = 230[70, 700] milliseconds
- Eye movement = 30 milliseconds
- Perceptual Processor = 100[50, 200] milliseconds
- Cognitive Processor = 70[25, 170] milliseconds
- Motor Processor = 70[30, 100] milliseconds

2.3.3.2 Cognitive-Perceptual-Motor GOMS (CPM-GOMS)

CPM-GOMS is known also as Critical-Path-Method GOMS. Bonnie John developed it in 1988 (Gray et. al, 1992). It differs from other GOMS variations, where it does not assume that user's interaction is a serial process. In other words, it assumes that perceptual, cognitive and motor operators can be performed in parallel.

However, tasks are broken down just to the level where they are still perceptual or motor. The tasks are first joined together serially and then examined to see which actions can overlap in parallel. This technique facilitates representation of overlapping. The estimated time by CPM-GOMS is generally faster since it does not allocate much time for the "prepare for action" type operations.

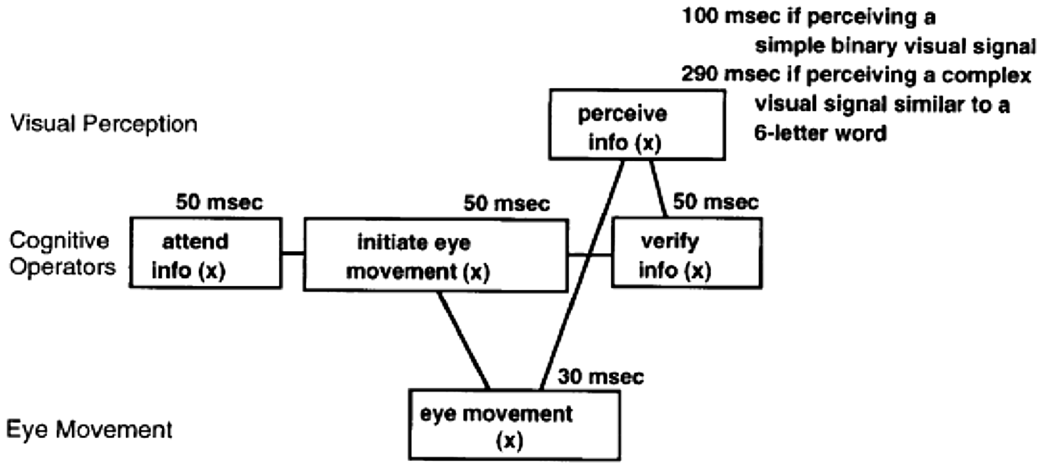


Fig 2.20 An example of a CPM-GOMS model (John and Kieras, 1996)

The figure 2.20 is an example of a scheduled chart for implementing the goal READ-SCREEN, when an eye movement is required. The sequence that produces the longest path through the chart is called the critical path. It represents an estimate of total time required to perform the task. In this case, the critical path is 50+50+30+100+50=280 msec. This particular model assumes that visual perception, cognitive operations, and eye movements can occur in parallel.

2.3.3.3 Natural GOMS Language (NGOMSL)

NGOMSL builds on CMN-GOMS by providing a natural-language notion for representing GOMS models. In addition, it is a procedure for constructing the models (John, 1990). Under NGOMSL, methods are represented in terms of an underlying cognitive theory known as cognitive complexity theory. This cognitive theory allows NGOMSL to incorporate internal operators such as manipulating working memory information or setting up sub-goals. Because of this, NGOMSL can also be used to estimate the time required to learn how to achieve tasks.

2.3.3.4 Keystroke-Level Model (KLM-GOMS)

KLM-GOMS is a simplified version of GOMS. It was developed in 1983 (Card et. al, 1983). The model is an 11-step method that can be used to estimate the time taken to complete simple data input tasks. KLM-GOMS is usually applied in situations that require minimal amounts of work and interaction with a computer interface or software design. Using KLM, execution time is estimated by listing the sequence operators and then summing the times of the individual operators. The original KLM had six classes of operators:

- K for pressing a key.
- P for pointing to a location on screen with the mouse.
- H for moving hands to home position on the keyboard.
- M for mentally preparing to perform an action.
- T (n) for type string of characters, n = number of characters.
- R for user waiting for the system to respond.

Each operation is assigned a duration as shown in table 2.2. It is intended to model the average amount of time an experienced user would take to perform the operation. Additionally, mental operations are placed by a set of rules that require some interpretation such as determining a conceptual "cognitive unit" or grouping of actions. They are inserted before each cognitive unit to account for cognitive preparation and decision-making.

Table 2.2 Original KLM operator’s duration (Kieras, 1993, 2001)

Classes	Duration	
K	0.08 seconds	Best Typist (135 wpm)
	0.12 seconds	Good Typist (90 wpm)
	0.28 seconds	Poor Typist (40 wpm)
	0.20 seconds	Average Skilled Typist (55 wpm)
	0.28 seconds	Average Non-secretary Typist (40 wpm)
	0.50 seconds	Typing Random Letters
	0.75 seconds	Typing Complex Codes
	1.20 seconds	Worst Typist (unfamiliar with keyboard)
P	1.10 seconds	
B	0.10 seconds	
H	0.40 seconds	
M	1.35 seconds	
T (n)	(n * K) seconds	

Being very fast is the main reason behind using this technique. It used to compare the speed of two different interfaces designed to accomplish the same task. On the other hand, a major caution is that the algorithm is designed to estimate the execution time for an expert user, which is typically faster than the time for a new user or an unfamiliar user.

Assuming error-free expert user interaction, KLM proved to estimate precise results, which is a drawback of similar models (Myung, R., 2004). Moreover, when the estimated experimental studies results were considerably off the actual values, the estimated difference between two examined designs still proved to be a strong basis for making design choices.

2.3.4 KLM for Phone Users

Limitation of research on performance measures for phone to the input of text for short messages was obvious in the past years. Dunlop and Crossan (2000) were of the first ones to show KLM operator sequences for three different text entry methods (traditional, predictive, and word completion). Despite the adoption of the original operator values used for desktop interaction, it was imprecise in this new mobile phones environment. Silfverberg et al. (2000) presented a model to predict expert text entry rates on a physical 12-key keypad mobile phone for more than one input method (multi-press, two-key, and T9). Predictions were provided for one-handed thumb and two-handed index finger input. They used Fitts' law and digraph probabilities to model the movement and linguistic components respectively. Table 2.3 shows the prediction equation for each input mode.

Table 2.3 Regression equations, R^2 values for Index and Thumb Fingers (Silfverberg et al., 2000).

Input Mode	Index Finger	Thumb
Regression equation	MT = 165 + 52 log2 (A/W + 1)	MT = 176 + 64 log2 (A/W + 1)
R^2	0.960	0.970

Mori et al. (2003) studied how the time values of the original KLM operators apply to mobile phone menu navigation and concluded that the operator values fit quite well with only minor modifications. In a complementary approach pursued by Pavlovych and Stuerzlinger (2004), non-perfect users are considered using the cognitive load operator to model input verification. They also compared several existing variants of the KLM and evaluated them with actual times from user studies. The authors concluded that the models give only rough approximations of real user behavior for text input. However, the models can correctly predict which input methods are faster than others (e.g., predictive over multi-tap).

More research was done on time measurements for key presses and mental act operator for different text entry languages, an example of which is Myung (2004) for Korean language. Furthermore, How and Kan (2005) defined 13 operators that directly map the phone keyboard interface according to the different input methods. They used videotaped sessions with a small set of subjects and a message-typing task in order to gather new times for these operators.

2.3.5 KLM for Mobile Device Interactions

The main use of mobile phones was to make phone calls, send text messages, and sometimes perform basic calendar tasks. Phones used very small displays and small physical buttons as the prime interface. With modern smartphones these buttons have been replaced with larger touch screens. Moreover, other uses are becoming more and more popular such as taking pictures, surfing the web, social networking, and playing music, videos and games. This adds several new interaction styles that have not yet been treated by interaction models.

There has been surprisingly little published research that includes new interaction techniques such as swipe, zoom, or tapping, for mobiles and tablets. Luo and John (2005) followed-up by Teo and

John (2006) were the first, as they showed that the method could soundly apply to handheld devices using stylus-based interfaces. They also presented a tool that automatically generates KLM models from storyboard descriptions. Holleis et al. (2007) adopted and defined a set of operators giving sound and study-based estimates of performance measures for each of them. They assumed that developers of mobile applications could then describe tasks as a sequence of the operators they added and predict user interaction times without needing to create prototypes. As part of their work on the user performance of multi-touch gestures on mobile devices, Tran et al. (2013) conducted an exploratory study of pinch and spread gestures with seated participants on a phone and a tablet device. They used Fitts' Law in order to characterize execution time for pinch and spread gestures on mobile devices (57% of the original data was used in the analysis). Each operation is assigned a prediction equation as shown in table 2.4.

Table 2.4 Regression equations, R^2 values for pinch and spread gestures (Tran et al., 2013).

	Phone		Tablet	
	Pinch	Spread	Pinch	Spread
Regression equation	MT = 360 + 304 log2 (A/W + 1)	MT = 295 + 323 log2 (A/W + 1)	MT = 260 + 317 log2 (A/W + 1)	MT = 231 + 337 log2 (A/W + 1)
R^2	0.945	0.927	0.964	0.884

Their spread gesture on tablet is equivalent to the Zoom interaction mentioned in chapter four. By comparing the results, shown in the developing and evaluation section, the outcomes are largely in line with slightly different prediction equation and higher correlating MT with ID (R^2 Zoom > 0.884).

2.4 Summary

The gap in research was the motivation behind introducing some improvements, which enable the users to be capable of carrying the text entry process despite their lack of experience dealing with the current keyboard layouts located on mobile devices and tablets. That is done by optimizing the Arabic keyboard layout on three design metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for less technologically literature users through approximate alphabetic sorting.

Many techniques for text entry on small displays have been proposed. However, there has been little research that merged gesture recognition with languages other than English. The proposed work focuses on Gesture-based text entry as a method of text entry for small screen mobile devices for Arabic language. In order to maximize the text area field and minimize the number of keys displayed on the screen of mobile devices, a 12-key GBAWP interface was introduced appearing like a 12-key physical keypad phone. Moreover, considering the Arabic letters characteristics, structure, and maximizing speed, a 6-key GBAWP layout based on dot recognition was introduced.

Enhancements to KLM have been proposed in the literature to be able to evaluate different techniques. However, there has been little research that improves user behavior modeling techniques that are capable of estimating the time taken to achieve common interactions performed on mobile touchscreen devices. The proposed work focuses on enhancing KLM by extending it with three new operators. An investigation is done on modeling these interactions performed on mobile devices and tablets. The proposed model is based on suggested times derived from Fitts' Law modeling and analysis of how well these interactions fit Fitts' Law. It also gives an enhancement for developers of smart-phone and tablet applications to predict user interaction times without even needing to create prototypes.

The main focus behind this chapter was to introduce HCI in general, and mobile HCI, UCD methodology, text entry concepts, and GOMS model in particular. The next chapter will show a study improving form control and navigation on mobile devices. It will give a brief description of an observation study conducted on the Egyptian Agricultural Census (EAC).

3. Improving Form Control and Navigation on Mobile Devices

In this chapter a study was done to improve form control and navigation on mobile devices. It is divided into three sections. The first section gives a quick review on literature. This quick review covers form fill-in concepts and techniques used to control and navigate the digital forms. In the second section, the improvements achieved through the study are introduced. Finally, a brief summary is presented.

3.1 Form Fill-in Concept

A form is a document (printed or electronic) with spaces to write or enter data. Forms are used to complete a certain task that has the requested information or question parts in common. It is used to increase the uniformity and efficiency of handling routine tasks, to ensure that tasks are completed, and to make sure that the information can be found when needed (Dayao, et al., 1990). People frequently have to fill out forms, and some with certain careers have to fill out the same form, over and over. For example, a salesperson may have to fill out a purchase order every time his client makes an order. A counter may have to fill out forms in order to conduct a certain type of census.

There have been sufficient researches that address the conversion of printed forms to digital ones. It sought the quality of interface and the ease of interaction. The quality of digital form interfaces depends primarily upon three factors. One of these factors is the clarity of the design and visual representation of the screens (Buxton & Baecker, 1987). This was due to the request of handling information spaces to fit a single screen. Nowadays, these single screens are becoming smaller and mobile such as; tablets and smart-phones.

Furthermore, in the bulk of current form design practice the element, layout, and functionality of user interfaces are intended for users who access the forms on a regular basis and prefer simple interfaces rather than more powerful alternatives. This has led to the adoption of basic point and click interaction model in which navigation and control were accomplished by selecting user interface elements with a pointing device such as the mouse (Ballard, et al., 2006). Today, new techniques are widely used to interact with touchscreen interfaces, particularly on mobile devices, yet the interaction model for forms has not been re-evaluated.

3.1.1 Form Navigation and Control Techniques

Navigation techniques provide a way to access vast information spaces through limited screen space. Scrolling (or panning) and zooming are fundamental techniques for freely moving around a two-dimensional continuous space. Scrolling allows the user to move to different locations, while zooming enables the user to view a target at different scales.

Completing complex forms is both time consuming and error prone with users making incorrect selections and spending considerable time navigating the form. Traditional scrollbars were used in navigating through large forms so that the large information space could fit on screen. Igarashi and Hinckley have identified that one of the major two limitations with using traditional interaction is that the users have to shift their focus between the document and the scrollbar (Igarashi & Hinckley, 2000). They suggest that this may increase the operational time and may cause a significant attentional overhead. Furthermore, in large documents small scrollbar movements can cause a large movement of the document that disorients the users.

However, there was a need to provide new navigation techniques so as to be used in complex form filling. To counter this visual blurring and to reduce physical navigational workload, there were two techniques introduced: Panning and Zooming vs. Tabbed navigation technique. First, Tabbed navigation technique, converts the form into several separate parts. Second, Pan and Zoom navigation technique, typical of image viewing applications, views the whole document as one navigable screen (Perlin & Fox, 1993; Bederson, et al., 1994).

3.1.1.1 ***Panning and Zooming Navigation Technique***

Panning and Zooming navigation technique used for document and map navigation tasks has proved to attain the user’s acceptance. It even added up to the improvement of the overall performance and proved to better perform than systems based on scrolling only (Kaptelinin, 1995). Nowadays, this technique is widespread on tablets and phones.

The objects are spatially organized in an infinite two-dimensional information space, and the user accesses a target object using Panning and Zooming operations. Successive zooming in and out by the user gradually expands and contracts the displayed contents while Panning changes the displayed section. In Zoomable User Interfaces such as; Pad (Perlin & Fox, 1993) and Pad++ (Bederson, et al., 1994), Panning and Zooming are coupled with the semantic zooming concept: the representation of an object depends on the scale of the information space. Moreover, a change in the size, shape, and details of the objects takes place while zooming not only but also appearance/disappearance from the visualization occurs.

However, in order to simplify the user's interaction, control menus solution was one of the proposed solutions for desktop systems (Pook, et al., 2000). It was introduced to provide unified and rapid selection of operations in a way that is similar to pie menus. Also, it allows control of the chosen operation in the same gesture. Additionally, various interaction techniques have been proposed to simplify scrolling, Panning and Zooming on mobile devices.

Rosenbaum and Schumann proposed an adaptation of the ZoneZoom technique to PDAs in 2005, which enables the users to easily explore large images on Smartphones. Each image is partitioned into nine cells; each one mapped into a number of the phone keypad, and pressing a key produces an automated pan and zoom on the associated cell (which can then be recursively partitioned into nine more cells). It functions by interacting with a grid overlaid on the currently displayed image portion through Panning and Zooming on images. The grid size is proportional to the size of the whole image and each grid cell can be tapped to zoom on the corresponding portion of the image. Merging or splitting the cells did provide the users with different zoom levels.

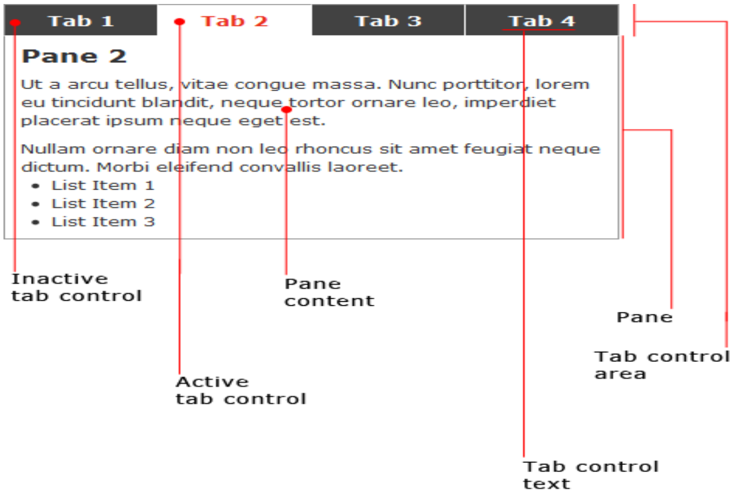
Jones et al. covered out an adaption to mobile devices in 2005, which was based on the Speed-Dependent Automatic Zooming (SDAZ) technique for navigating documents proposed by Igarashi and Hinckley in 2000. SDAZ operates by combining scrolling and zooming into one operation, where the zoom level is inversely proportional with the scroll speed. This technique outperformed standard scroll, pan and zoom methods in document and map navigation tasks (Cockburn & Savage, 2003). The SDAZ, version by Jones et al. (2005), operates as follow:

- Two concentric circles are drawn when users tap on the information space with the pointer.
- If the pointer remains within the inner circle, the user is free to pan in any direction and the panning rate increases as the pointer moves away from the starting position.
- When the pointer moves beyond the inner circle, both zooming and panning operations take place.
- The information space progressively zooms out as the user moves closer to the outer circle and the panning speed changes to maintain a consistent visual flow.
- When the pointer reaches the outer circle, no further zooming occurs, while panning remains active.

*3.1.1.2 **Tabbed Navigation Technique***

Tabbed navigation is a User Interface (UI) design pattern where content is separated into different panes where each pane is viewed one at a time. The user requests content to be displayed by clicking the content's corresponding tab control. It is one of the most enduring techniques used in websites and web applications leading to optimizing page screen areas without sacrificing the amount of information presented at once, and have become conventional enough to be understood by almost all users. Tabs generally tie the navigation to the content visually. The result is that the user will instantly recognize that by using this navigation technique, the displayed content will be affected.

As shown in figure 3.1, the key parts of the tabbed navigation technique will be identified and a brief description of each part will be given.



Pane 2

Ut a arcu tellus, vitae congue massa. Nunc porttitor, lorem eu tincidunt blandit, neque tortor ornare leo, imperdiet placerat ipsum neque eget est.

Nullam ornare diam non leo rhoncus sit amet feugiat neque dictum. Morbi eleifend convallis laoreet.

- List Item 1
- List Item 2
- List Item 3

Fig 3.1 Tabbed navigation technique

- The tab control area is the location of the tab controls; these controls are the interface component for navigating through panes of tabs.
- The tab control text is the text that describes the tab control. It should be short (one to two words) and should effectively depict the corresponding pane information.
- The active tab control refers to the tab control that is presently selected. Only one tab control should be active. The first tab control is the default active tab.
- Inactive tab controls are the tab controls whose panes are not currently showing.
- A pane is where information is displayed; it should have a corresponding tab control so that panes that are not displayed are accessible by clicking its tab control.
- Pane content is presented inside a pane.
- The active pane is the pane that is currently being shown; it is paired with the active tab control. The pane that is displayed immediately is the default active pane.
- Inactive panes are the panes that are currently not being shown. An inactive pane becomes the active one when its tab control is clicked.

The primary goal of the tabbed navigating UI pattern is to permit users to view a group of related data one at a time, which in turn allows designers to modularize this group of information in a compacted manner, saving valuable screen real estate.

The discussion so far covered the concept of form fill-in and techniques used to control and navigate digital forms. The next section will present an observation study on the EAC and the steps used to improve the form control and navigation on mobile devices.

3.2 Improved Form Control and Navigation on Mobile Devices

In this section, the work done on improving form control and navigation on mobile devices will be introduced. The first subsection will focus on the observation study conducted on the Egyptian Agricultural Census (EAC). The second subsection will show the development of electronic form used in EAC. The final subsection will discuss testing the forms and showing the results.

3.2.1 The Observational Study

The EAC deals with an inventory of all possessions located in villages and cities of the Nile Valley, as well as possessions located in desert governorates and land belonging to the reconstruction organization.

The EAC is conducted through two phases. First phase, a sketch is set at the beginning for all villages and towns nationwide. Then the buildings are numbered and a visit is made to all the holders to collect data. Second phase, a visit is made to all the families to collect more precise data including personal information and other detailed data regarding their possession. All gathered data are written in paper forms. The preliminary stage of the EAC contains four steps; issuing executive decisions to conduct the EAC, preparing the census budget and adopting it, choosing the field supervisors, engineers, agents, and participants in the governorates, and finally training workers on the agricultural census. A trial census is done in order to measure the efficiency of the forms used, the method of data collection is revised, problems and obstacles are identified, and finally the response of holders to questions on the census are observed. This trial census was held in three selected governorates representing the regions of Upper and Lower Egypt.

The following flow chart illustrates how the processes of the census and data collection are carried out.

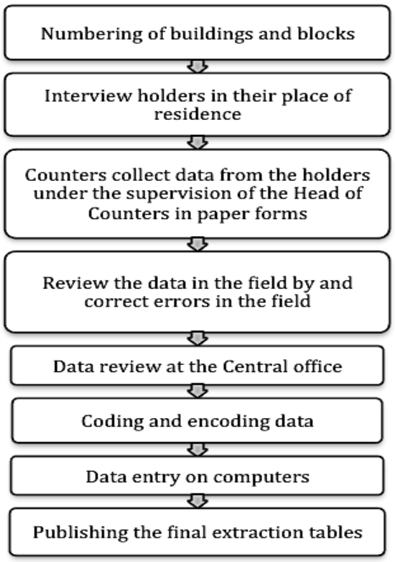

Numbering of buildings and blocks
Interview holders in their place of residence
Counters collect data from the holders under the supervision of the Head of Counters in paper forms
Review the data in the field by and correct errors in the field
Data review at the Central office
Coding and encoding data
Data entry on computers
Publishing the final extraction tables

Fig 3.2 Egyptian Agricultural Census Process Flow

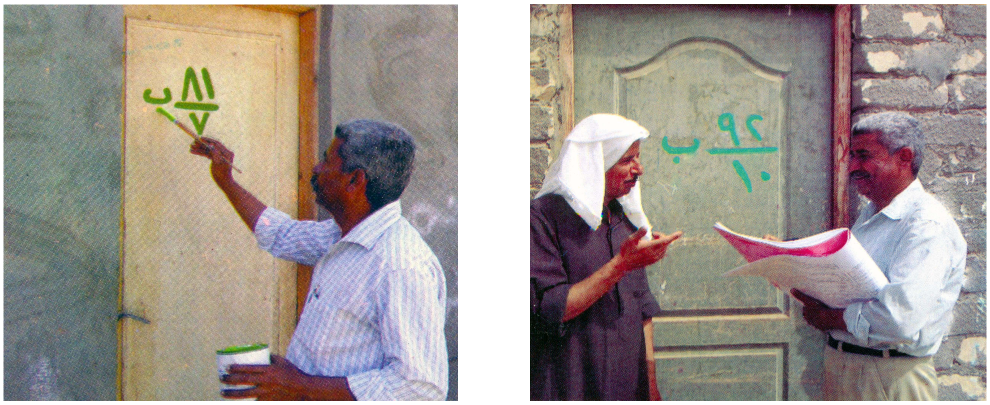


Fig 3.3 Build Numbering and Census interviews while completing paper forms

As shown in figure 3.3, the census is conducted through personal interviews done by counters with the holders in their place of residence. That took place after numbering the buildings in villages and small towns to identify them easily. However, due to the difficulties of numbering in big cities, the counters follow the method of inventory by walking through agricultural basins and taking landholders addresses in order to reach them.

There are certain obligatory rules, set by the Ministry of Agriculture, which must be adhered to by counters and the head-of-counters. First, the allocation of one head-of-counters to each five counters. Second, the equitable is taken into account distribution among counters and heads-of-counters in areas of operation. Third, the revision of the numbering that took place in the first phase, buildings not mentioned should be documented by writing a notice. Forth, buildings must be visited in the same order of the records and all the holders contained in the records must be visited. Finally, the heads-of-counters accept records after being reviewed precisely.

3.3 Development and Evaluation of the Electronic Form

After indicating the importance of involving EAC counters in the design and development processes, a study was conducted on enhancing the form navigation using Panning and Zooming on mobile devices and comparing usage against Tabbed navigation technique.

3.3.1 Participants

A number of 20 male EAC counters, nominated by and worked at the Egyptian Ministry of Agriculture, were recruited in Egypt. However, they were all familiar with the original paper version of the Agricultural Census. Their age ranged from 25 to 42. They were all bachelor degree holders. Their fields of study greatly varied and none of them were familiar with using tablets, while their usage for desktop/laptop was limited to home use entertainment such as; playing games and watching movies.

3.3.2 Apparatus

The form is designed[2] by using HTML5 markup and displayed on an Apple iPad 2 using the standard Safari browser (9.7" screen with resolution of 1024x768). It is based on two techniques; selecting and typing. The form is composed of a combination of checkboxes and textboxes and was designed to closely resemble the original paper version in structure and individual components

The paper Agricultural Census form is divided into 5 main sections. Section 1 contains questions regarding the holder's information. Section 2 data is regarding the possession of land. Section 3 is data about livestock and animals. Section 4 discusses the agricultural machinery. The final section includes information regarding labor used. These five sections are reflected in the electronic form. The first section is visible to enter in the data. The other four are optional and initially hidden and become visible according to the selection of their checkbox on the overview section.

As shown in figure 3.4, the counter enters the essential data of the holder regarding his/her personal information. This part represents the first section of the form. The counter asks the holder 28 yes/no questions as shown in figure 3.5; these questions generate the entire form. There are four main questions representing the rest four main sections of the form. Each main question contains a number of sub questions, which represent a set of required information to be given by the holder and filled out by the user (counter).

[2] The study systems were developed in HTML and JavaScript by the candidate.

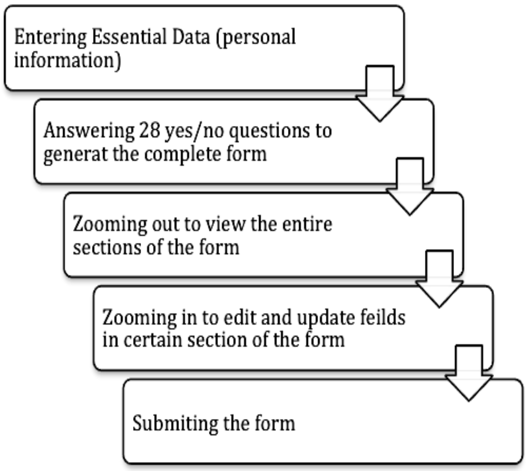
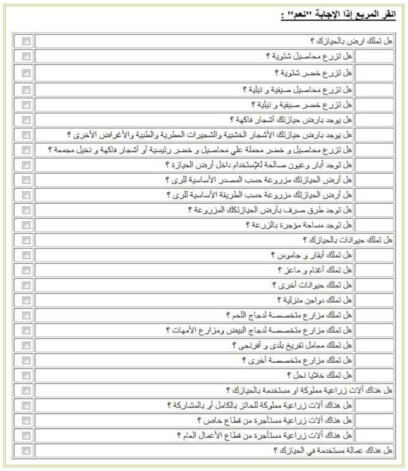

Entering Essential Data (personal information)
Answering 28 yes/no questions to generat the complete form
Zooming out to view the entire sections of the form
Zooming in to edit and update feilds in certain section of the form
Submiting the form

Fig 3.4 Form Filling Process Flow Chart



انقر المربع إذا الإجابة "نعم" :

☐ هل تملك ارض بالحيازك ؟
☐ هل تزرع محاصيل شتوية ؟
☐ هل تزرع خضر شتوية ؟
☐ هل تزرع محاصيل صيفية و نيلية ؟
☐ هل تزرع خضر صيفية و نيلية ؟
☐ هل يوجد بارض حيازتك أشجار فاكهة ؟
☐ هل يوجد بارض حيازتك الأشجار الخشبية والشجيرات العطرية والطبية والأغراض الأخرى ؟
☐ هل تزرع محاصيل و خضر محملة علي محاصيل و خضر رئيسية أو أشجار فاكهة و نخيل مجمعة ؟
☐ هل توجد آبار وعيون صالحة للإستخدام داخل أرض الحيازة ؟
☐ هل أرض الحيازتك مزروعة حسب المصدر الأساسية للرى ؟
☐ هل أرض الحيازتك مزروعة حسب الطريقة الأساسية للرى ؟
☐ هل توجد طرق صرف بأرض الحيازتكك المزروعة ؟
☐ هل توجد مساحة مؤجرة بالزرعة ؟
☐ هل تملك حيوانات بالحيازك ؟
☐ هل تملك أبقار و جاموس ؟
☐ هل تملك أغنام و ماعز ؟
☐ هل تملك حيوانات أخرى ؟
☐ هل تملك دواجن منزلية ؟
☐ هل تملك مزارع متخصصة لدجاج اللحم ؟
☐ هل تملك مزارع متخصصة لدجاج البيض ومزارع الأمهات ؟
☐ هل تملك معامل تفريخ بلدى و أفرنجى ؟
☐ هل تملك مزارع متخصصة أخرى ؟
☐ هل تملك خلايا نحل ؟
☐ هل هناك آلات زراعية مملوكة او مستخدمة بالحيازك ؟
☐ هل هناك آلات زراعية مملوكة للحائز بالكامل أو بالمشاركة ؟
☐ هل هناك آلات زراعية مستأجرة من قطاع خاص ؟
☐ هل هناك آلات زراعية مستأجرة من قطاع الأعمال العام ؟
☐ هل هناك عمالة مستخدمة في الحيازك ؟

Fig 3.5 28 (yes/no) questions

Subsequently, zooming in or out enables the counter to edit and update fields in a certain section or to view the entire sections of the form, respectively. Afterwards, the counter begins filling in the form. Later, after making sure that all fields are completed and correct, the user submits the form.

3.3.3 Procedure and Results

Users were asked to complete two forms on the device. Figures 3.6 and 3.7 show the two interface versions. The dependent variables were task-completion time (in seconds), interface preference, and subjective workload. The workload was measured with the NASA Task Load Index questionnaire (NASATLX), which contains six subscales; Mental Demands, Physical Demands, Temporal Demands, Own Performance, Effort, and Frustration. Tasks and order of presentation of interfaces were randomly balanced.

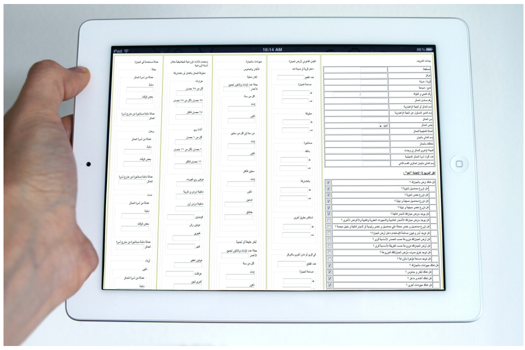


Fig 3.6 View of the entire form using Panning and Zooming navigation technique

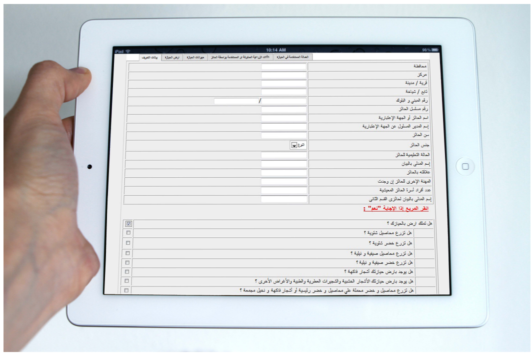


Fig 3.7 View of the form using Tabbed navigation technique

The results showed that the participants took about 31 minutes on average to complete the tasks using the Panning and Zooming technique, but 35 minutes on average using the Tabbed navigation technique. This significantly showed lower task completion time ($p<0.05$, paired t-test, df =19) as shown in figure 3.8. Moreover, NASA TLX forms were completed after each interface and showed lower workload for effort ($p<0.1$) using Panning and Zooming navigation technique. This implies that the users did low effort in order to accomplish their level of performance.

Table 3.1 Time and NASA TLX results

	Completion time (min)	NASA TLX Scale					
	T	Q1	Q2	Q3	Q4	Q5	Q6
Zoom	31	10	9	12	13	9	12
Tab	35	12	11	11	11	11	12
T-Test results	2.163	0.719	1.044	0.968	0.653	1.871	0.689

Finally, at the end of the test, users were asked, “which navigation technique do you prefer?”. Fourteen of twenty users favored the Panning and Zooming technique. Their comments were clear regarding the ability of viewing the whole form rather than hopping through tabs. A couple of them suggested adding the ability of the user to determine the uncompleted sections or fields in the form.

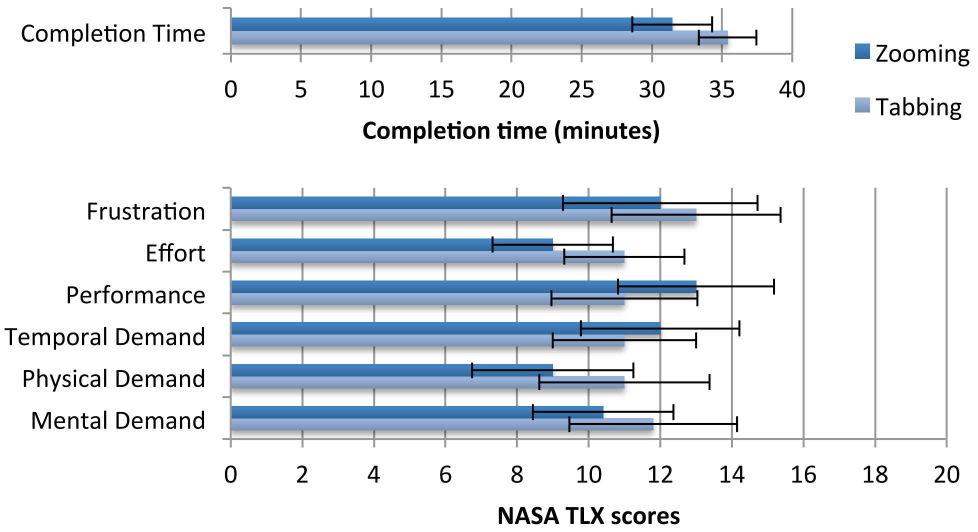


Fig 3.8 Time and NASA TLX results

3.4 Summary

This chapter shows an improvement of form navigation on mobile devices. The approach is based on computerizing forms and using Panning and Zooming as a navigation technique. In order to do so, an observational study was conducted on the Egyptian Agricultural Census (EAC). However, there were considerable challenges in reducing the size of the paper forms to fit mobile devices and introducing fast navigation technique.

It was concluded that using Panning and Zooming technique scored 4 minutes less in the average completion tasks time in comparison to using Tabbed navigation technique. Eventually, by measuring the workload using the Panning and Zooming technique, it showed indicatively lower workload which reinforces its superiority over using Tabbed navigation technique in form navigation. The results are based on a relatively small sample of participants doing artificial tasks in a laboratory setting. Future work could be conducted to run field-based trials as part of the EAC. This is to confirm and fully assess the results of the zoom-based tablet application that supersedes the paper-based forms.

The next chapter will illustrate the optimization of Arabic text entry on mobile devices. It will give a brief description of an informal survey on key distribution of Arabic keyboard and the way of optimizing the Arabic keyboard layout.

4. Optimizing Arabic Text Entry on Mobile Devices

There are many improvements proposed in the field of HCI and especially in text entry in which the Arabic Language users was involved in the enhancement process. The research aims at implementing these improvements, which enable the Arabic Language users to carry out the data entry process efficiently using the current keyboard layouts located on mobile devices and tablets. That is done by optimizing the Arabic keyboard layout on three design metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for less technologically literature users through approximate alphabetic sorting.

In this chapter, the work done on optimizing the Arabic keyboard layout will be shown. The first subsection will cover an informal survey on key distribution of Arabic keyboard. The second subsection will focus on the development of an optimum keyboard. The third subsection will show the testing of the developed keyboard and the outcomes of the tests. The final subsection will give a brief summary.

4.1 Informal Survey on Key Distribution of Arabic Keyboard

After indicating the lack of research regarding optimizing the Arabic keyboard layout, an informal survey was conducted on key distribution of Arabic keyboard. In this survey vertically aligned alphabetically ordered keyboard layout was compared to the default Arabic keyboard layout. The main goal was to know which keyboard the Arabic Language users prefer.

4.1.1 Participants

Ten Egyptian volunteers helped in conducting the survey. Their main speaking language was Arabic. Five aged 26-35, four aged 36-49, one aged 50, and 30% females and 70% males. All participants were bachelor degree holders. Their fields of study and work varied greatly. Only one participant was familiar with touch screens. Additionally, four participants were desktop/laptop users.

4.1.2 Apparatus

Two keyboards shown in figures 4.1 and 4.2 were created as paper prototypes. The default Arabic keyboard layout prototype was captured from an iPhone 4, while the second keyboard was designed in a way such that the keys are aligned vertically from right to left in alphabetical order.

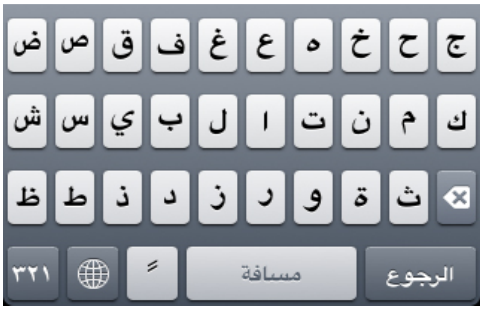


Fig 4.1 Default Arabic keyboard layout

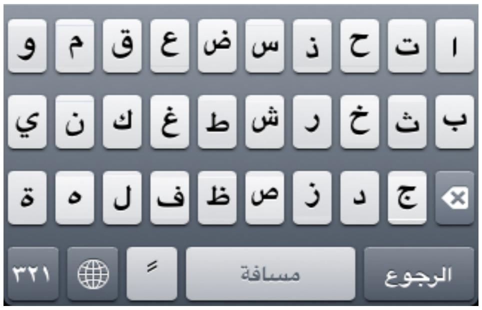


Fig 4.2 Vertically aligned alphabetically ordered keyboard layout

4.1.3 Procedure and Results

Participants were asked to type a couple of random sentences and to choose which layout is preferable from their point of view. Six out of ten preferred the keyboard with letters aligned vertically in alphabetical order, due to the ease of finding the letters. Three preferred the keyboard with the default Arabic layout because they are accustomed to using it. One was neutral and wanted to align the letters in alphabetical order but horizontally from top to bottom.

4.2 Development of an Optimized Arabic Keyboard Layout

After going through the findings from the survey regarding which keyboard is preferable, it was found that the default Arabic keyboard layout is not preferable to Arabic Language users. Moreover, introducing a keyboard with alphabetically sorting layout that maximizes text entry speed and minimizes error occurrence is required. However, in order to achieve this optimal keyboard layout, it was necessary to go through a certain methodology. This methodology depended on using an optimization technique and a large representative Arabic text corpus.

Pareto optimization is a variant of local neighborhood search adapted for multiple criteria searching (Hoos, H.H. and Stutzle, T, 2005). In other words, it is an area of multiple criteria decision making, where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. A three-dimensional Pareto front optimization (Dunlop, M. D. and Levine, J, 2012) was used to minimize finger travel distance in order to maximize speed, minimize neighboring key error ambiguities in order to maximize the quality of spell correction, and maximize familiarity for less technologically literate users through approximate alphabetical sorting.

4.2.1 Arabic Text Corpus

The development of a keyboard requires analysis of a large representative text corpus. Issues of the Arabic newsletter “Watan” dated back to 2004 were analyzed. It contained articles from fields such as culture, economy, international and regional news, religion, and sports (2782, 3468, 5631, 3860, 4550 articles respectively). So the related character frequencies are not biased to any particular field. The articles were downloaded, parsed and analyzed to find frequency information.

The articles were extracted, saved in one file (size of 72 MB), and all numbers and symbols were removed. Occurrence of characters, character-pairs and badgrams were calculated. Badgrams were based on the likelihood of a one-letter substitution resulting in a valid word and, as per Dunlop and Levine, were weighted according to frequency of use. The corpus contained 126,913 unique words (34,009,607 occurrences).

4.2.2 Monograms, Bigrams, and Badgrams

Monogram from the Arabic developed corpus is shown in figure 4.3. The x-axis represents the letters of the Arabic language, while the y-axis in represents the monogram scaled to the total number of Arabic language letters. The results show that letter ‘ا’ recorded the highest frequency among all letters, which implies that it is the most used letter in the Arabic language with a frequency of 0.187. The second, third, and last frequencies were ل = 0.111, ي = 0.099, and ظ = 0.001 respectively. These frequencies will help in identifying the new position of letters in the optimization process.

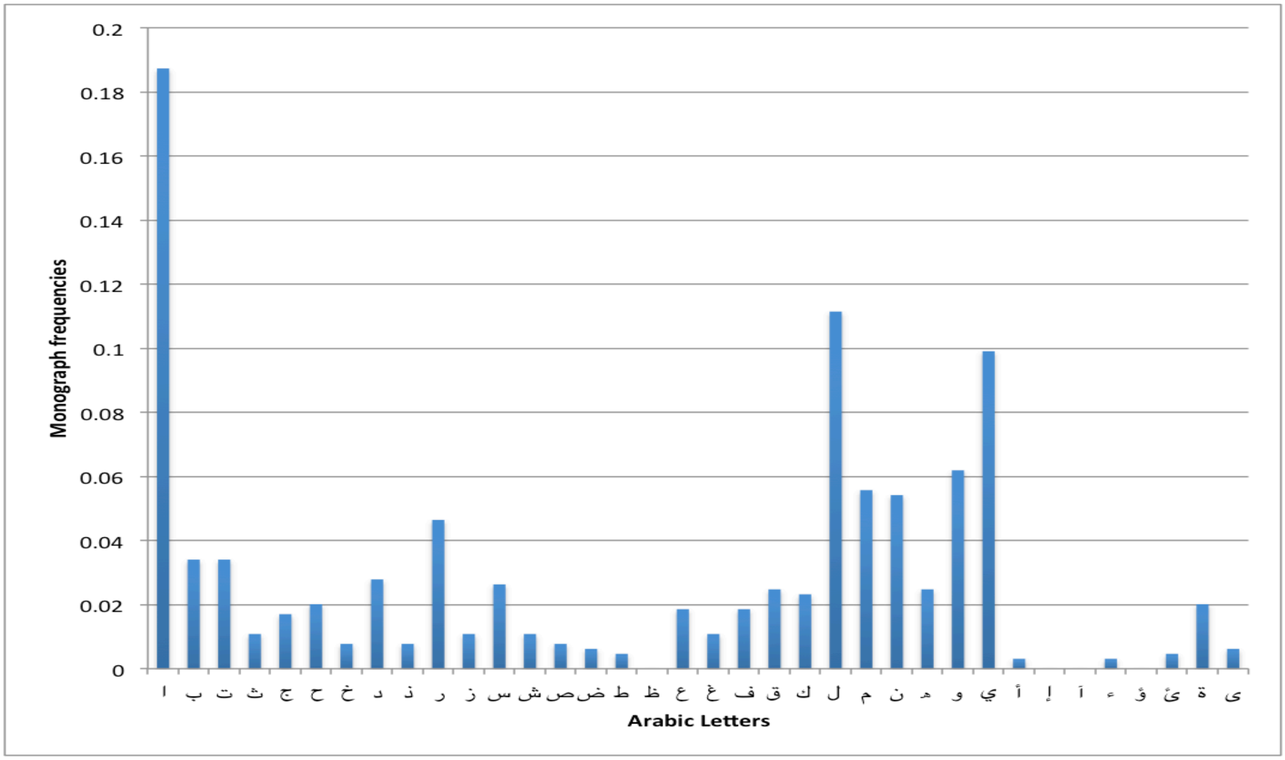


Fig 4.3 Arabic letter monogram frequencies in the Arabic corpus

The bigram from the Arabic developed corpus is shown in figure 4.4. The x-axis and z-axis represent the occurrence of two consequent letters, while the y-axis represents the bigram scaled to the total number of letters. The bigram analysis resulted in a weight for each two-letter Arabic bigram – the occurrence information between the 36 characters and between each character was recorded and space to give 37*37 pairs. The top four key pair probabilities were ي_ = 0.071, ن_ = 0.068, _ا = 0.060, and ال = 0.054 (where _ represents space), as shown in table 4.1. Compared to English there were very few badgrams for Arabic. This is a result of the reduced role of vowels in written Arabic. The most common badgrams were "ام" and "يم", however 32 badgrams in total were found (compared to 325 for English), where "ام" represents only 0.003% of the bigram occurrences compared to 0.017% for "AE" in English.

Table 4.1 Arabic letter pair frequencies in the Arabic corpus.

	_	ا	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع
_	0.000	0.012	0.001	0.002	0.001	0.000	0.000	0.000	0.003	0.000	0.001	0.002	0.001	0.000	0.000	0.004	0.000	0.000	0.003
ا	0.060	0.000	0.001	0.000	0.002	0.002	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.003
ب	0.004	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0. 00
ت	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ث	0.002	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ج	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
ح	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
خ	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0 000	0.000
د	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0 000	0.002
ذ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0 0
ر	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
ز	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	000
س	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ش	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0 000	0.0 0
ص	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ض	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ط	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ظ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ع	0.020	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
غ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
ف	0.047	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0. 00	0.000
ق	0.001	0.001	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	. 0
ك	0.010	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	000	0.000
ل	0.007	0.054	0.001	0.000	0.000	0.001	0.003	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	.015
م	0.045	0.003	0.000	0.000	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0. 00	0.000
ن	0.000	0.014	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005
ه	0.009	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
و	0.004	0.007	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
ي	0.001	0.001	0.002	0.007	0.000	0.000	0.001	0.000	0.000	0.004	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
أ	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
إ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0 0	0.000
آ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00
ء	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0 00	0.0 0
ؤ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ئ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ة	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
ى	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 3.1 Arabic letter pair frequencies in the Arabic corpus (continued).

	غ	ف	ق	ك	ل	م	ن	ه	و	ي	أ	إ	آ	ء	ؤ	ئ	ة	ى
_	0.000	0.001	0.002	0.000	0.019	0.009	0.068	0.006	0.000	0.071	0.000	0.000	0.000	0.002	0.000	0.000	0.005	0.017
ا	0.000	0.001	0.001	0.005	0.011	0.004	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ب	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ت	0.000	0.000	0.000	0.000	0.007	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ث	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ج	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ح	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
خ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
د	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ذ	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ر	0.000	0.000	0.000	0.000	0.001	0.002	0.000	0.000	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ز	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
س	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ش	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ص	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ض	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ط	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ظ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ع	0.000	0.000	0.000	0.000	0.003	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
غ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ف	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ق	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ك	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ل	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
م	0.000	0.000	0.000	0.000	0.006	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ن	0.000	0.000	0.000	0.001	0.007	0.038	0.000	0.000	0.002	0.002	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ه	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
و	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ي	0.000	0.047	0.000	0.002	0.013	0.001	0.003	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
أ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
إ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
آ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ء	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ؤ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ئ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ة	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ى	0.000	0.000	0.000	0.000	0.015	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

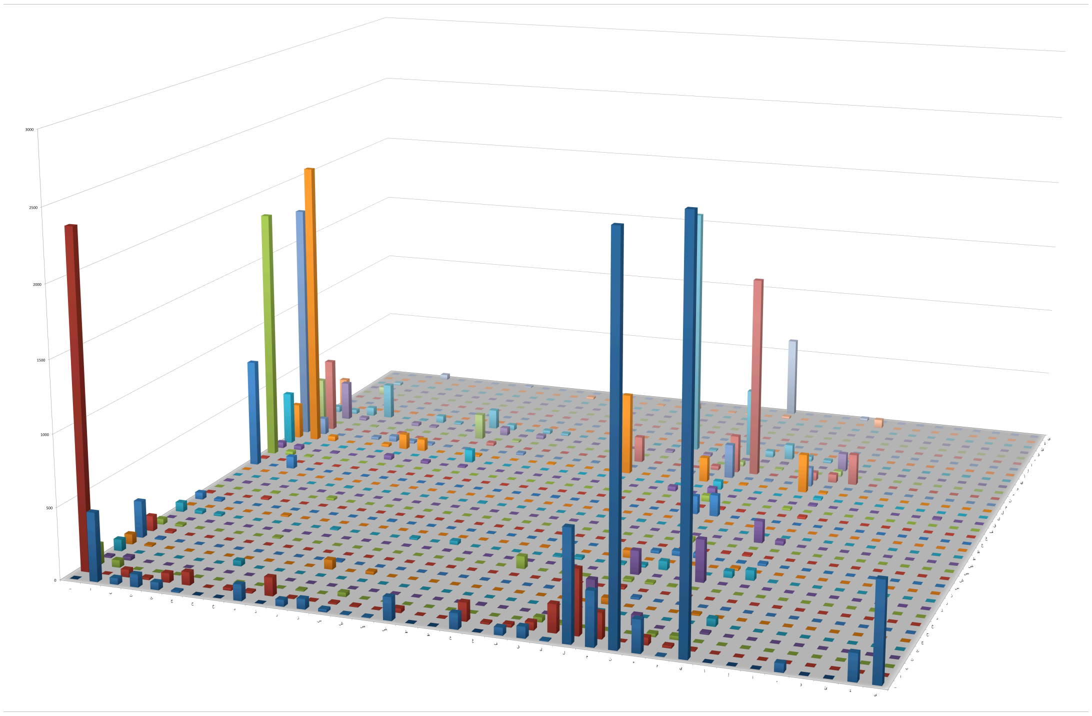


Fig 4.4 Arabic letter bigrams in the developed corpus

A function that measured similarity between novel layouts and QWERTY was added by Dunlop and Levine (2012) as a third metric to attempt to minimize deviation from QWERTY. Given the higher levels of computer and keyboard literacy in developing countries, such as Egypt, together with the variation in keyboard layouts, optimizing to a standard keyboard was found less suitable. Accordingly, the third metric was familiarity with a column-major alphabetic keyboard, with alphabets running from right to left.

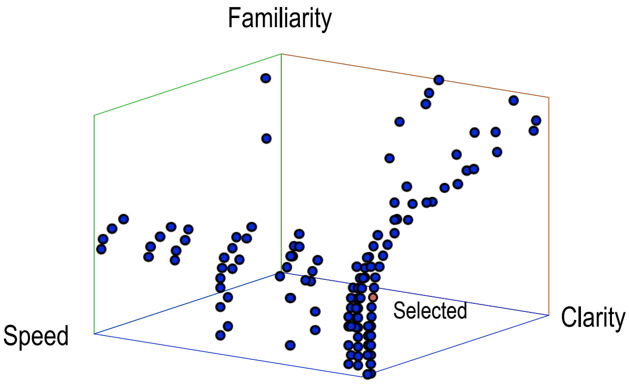


Fig 4.5 Pareto Front Shape

Randomly generated keyboards formed an initial set of potential solutions[3]. This set went through 2000 iterations in which each keyboard had a small number of keys swapped. If the new keyboard was better on any metric than an existing solution, it was added to the set. On the other hand, if it was good on all metrics then it dominates the existing one. This leads to a Pareto front – a set of dominant solutions on a 3D surface as shown in figure 4.5. At the end of the process the space was normalized so that each dimension ranged from 0 to 1 and took a good compromise design from the center of the Pareto front.

4.3 Evaluating the Optimized Arabic Keyboard

Two user studies were conducted in which the selected Pareto optimized Arabic keyboard layout was evaluated against the Apple Arabic layout.

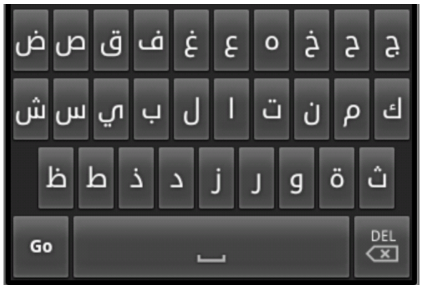


Fig 4.6 Apple Arabic keyboard layout

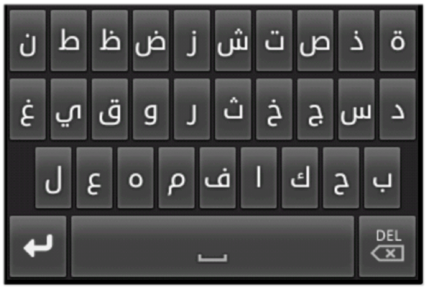


Fig 4.7 Pareto optimized Arabic keyboard layout

4.3.1 Participants

Twenty volunteer participants were recruited in Egypt by personal contact to conduct each user study (total of two user studies). Their main speaking language was Arabic. Participants were Egyptians, aged 20-40, 45% females and 55% males, and bachelor degree holders. Their fields of study and work varied greatly. Twelve participants were familiar with touch screens. Also, seventeen participants were desktop/laptop users.

[3] Based on previous code for English optimization, Dr. Mark Dunlop developed the Pareto optimized Arabic keyboard layout code and ran the software to generate the keyboard layouts. Corpus analysis, refinement, bigram and badgram modelling were done exclusively by the candidate. The Android IME Java and XML code was developed jointly.

4.3.2 Apparatus

Two keyboards shown in figures 4.6 and 4.7 were created. The developed keyboards were an Android implementation and displayed on an HTC Desire S (480 x 800 pixel, 3.7" screen size), running Android OS, v2.3 (Gingerbread).

4.3.3 Procedure

Users were asked to type in 3 full sentences as a practice and 20 full sentences as a test on each keyboard. Time information was recorded from the first keystroke to the last keystroke for each sentence. These sentences were selected randomly from the standard MacKenzie and Soukerof set developed short phrases for the study and translated into Arabic (MacKenzie, I. S. and Soukoreff, R. W., 2003), as shown in appendix D. Users were asked to correct any mistakes while typing in the sentences. They were told that typing speed and accuracy were the interest behind these two user studies.

4.3.4 Results and Discussion

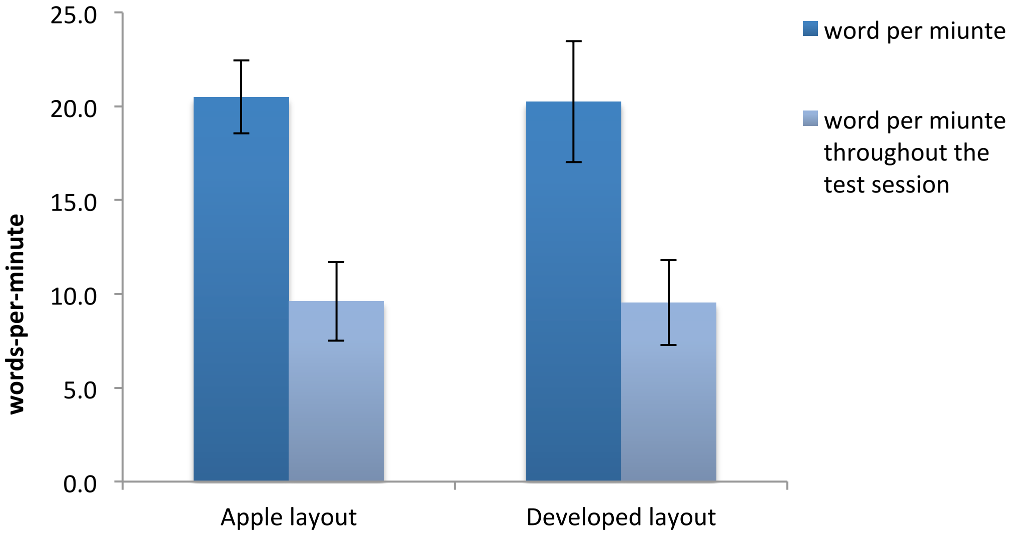


Fig 4.8 Text entry results of first usability test

The results in figure 4.8 showed that users typed at a rate of 20.3 and 20.5 words-per-minute on average using the Pareto optimized Arabic keyboard layout and the Apple Mac Arabic keyboard layout, respectively. It also showed that users typed at a rate of 9.5 and 9.6 words-per-minute throughout the test session on average using the Pareto optimized Arabic keyboard layout and the Apple Mac Arabic keyboard layout, respectively. The uses began their sessions using the Pareto optimized Arabic keyboard layout, which increased the learning effect through the study. Consequently, these results in addition to the learning effect ensured the need to conduct another user study with another 20 volunteers, to make sure that the optimized Arabic keyboard could overcome the default Apple Arabic keyboard.

In the second user studies, ten uses began their sessions using the Pareto optimized Arabic keyboard layout, while the other ten began using the Apple Mac Arabic keyboard layout. The results in figure 4.9 showed that users typed at an estimate of 24.4 words-per-minute[4] on average using the Pareto optimized Arabic keyboard layout, but 19.6 words-per-minute on average using the Apple Mac Arabic keyboard layout. Despite the relatively short study, this shows an indicative speed advantage for the developed keyboard (2-tailed paired t-test, n=20, p<0.1) over the standard keyboard. It also showed that users typed at a rate of 11.4 words-per-minute throughout the test session on average using the Pareto optimized Arabic keyboard layout, but 9.2 words-per-minute throughout the test session on average using the Apple Mac Arabic keyboard layout.

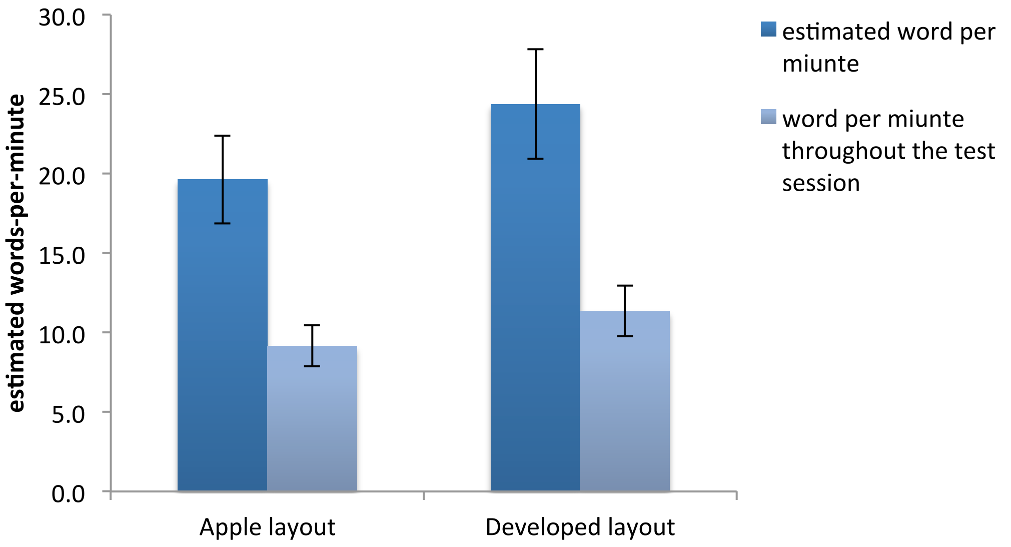


Fig 4.9 Text entry results of second usability test

[4] The system failed to record wpm data from some phones in the second usability study. Times for these were estimated based on server logs and estimates crosschecked where full data was available.

Throughout the study, 13 users performed faster on the Pareto optimized Arabic keyboard, whereas 4 accomplished the testing task quicker using the Apple Mac keyboard, and just one user typed at the same speed on both layouts. At the end of the study, users were asked to select their most preferred keyboard layout. The majority preferred the Pareto optimized Arabic keyboard layout. Their main preference was due to the key distribution being close to the vertically aligned alphabetically ordered keyboard layout, which enabled them to find keys faster. However, some users noted that more time is needed to practice on the optimized keyboard layout, to shift from the Apple Mac layout to the optimized one.

4.4 Summary

This chapter presents a new design of an Arabic keyboard layout for effective text entry on touch screen mobile phones. The approach is based on Pareto front optimization using three metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for Arabic Language users through approximate alphabetic sorting.

In user studies, the new layout showed an observed improvement in typing speed in comparison to the Apple Mac Arabic Keyboard layout. In addition to this initial benefit, the design of the keyboard to support faster and more accurate entry should lead to even higher medium and long-term speed. Users' preferences were obtained in favor of the optimized layout, as key distribution is more familiar to the vertically aligned alphabetically ordered layout. Furthermore, comparing Arabic Language with typical Western users, having a wider variation in standard keyboard layouts, some users still suggested that it would take more time practicing to gain more familiarity with key locations.

Currently, there is an opportunity to research new optimized keyboard designs with less usage experience than QWERTY as in mainstream Western European languages. Pareto optimization can produce high quality keyboards for alphabet based languages that could be beneficial when there is less reluctance to change from QWERTY.The next chapter will introduce a Gesture-based text entry technique on small display mobile devices.

5. Gesture-Based Arabic Writing Pad

Gesture-based text entry was introduced as a method of text entry for mobile devices. In this method, gestures are drawn instead of touching keys on the screen. Handwritten character can be a gesture, but instead of letters, symbols and metaphors are used such as; edges, points, swipes, and spaces. This method could hinder the user's writing on the screen. However, it proved its superiority when it comes to the recognition accuracy and speed because it guarantees a correct output when user writes a correct gesture on the screen.

In this chapter, the work done on introducing an Arabic Gesture-based writing pad will be shown. The first section will cover the concept and methodology behind introducing such writing pads. The second section will show a study on fragmentation text entry for Arabic Language. The third section will focus on the development and the test of a 12-Key GBAWP. The third section will illustrate the development and the test of an optimized 6-Key GBAWP suitable for small display mobile devices. The final section will give a brief summery.

5.1 Concept and Methodology of the GBAWP

A recent method known by Gesture Keyboard is one of the methods used for text input for syllabic scripts. Its success and acceptance is critically dependent on the reliability of gesture recognition. The gesture recognition for the GBAWP is accomplished through sequence of touched points or swipes on the screen. The architecture of GBAWP is shown in figure 5.1, were the activity of gesture recognition is divided across two major areas known respectively as the Screen and the Recognition Engine. The Screen recognizes the simple swipes done by the user and views back intended characters, while the Recognition Engine achieves swipe data from the Screen, assembles them to form sets, and converts the correct set into defined characters.

The process, as shown in figure 5.1, is initiated when user sets his/her finger on the screen of the device, which sets the touchStart function indicating the X/Y position of the finger. During the movement of the finger, touchMove function records the motion of the finger. Upon the removal of the user's finger from the screen, touchEnd function gathers all data from the previous two functions and sends it to the recognition engine in order to identify the intended character.

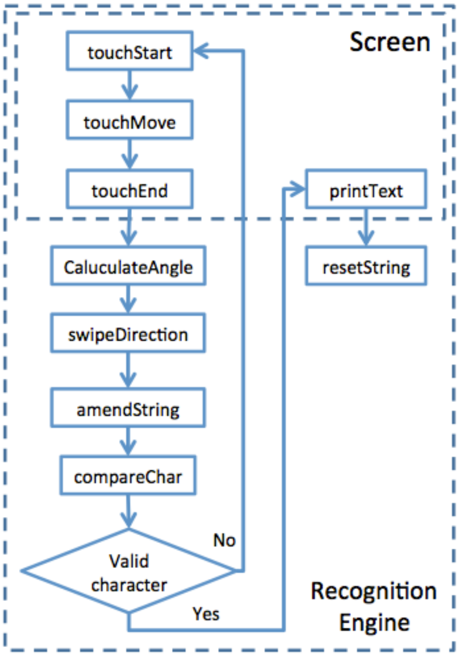


Fig 5.1 GBAWP Process Flow

Afterwards, two functions within the recognition engine receives the data, perform some calculations to identify the swipe direction. There are four direction used in GBAWP; North, South, East, and West. After detecting the swipe direction, it is amended to a string that is used afterward in the comparison process. Pre-defined sets are initiated in the recognition engine; each set defines a valid Arabic character. The compareChar function is responsible for comparing the string, which contains a set of swipes, with the pre-defined sets. If the string does not match a pre-defined set, the recognition engine rests till it receives new data from the screen. Otherwise, it converts the set into an Arabic character and sets the printText function to print out the character on the screen. Finally, a resetString function is initiated to reset the string to null on the engine's side waiting for new data.

5.2 A survey on fragmentation text entry for Arabic Language

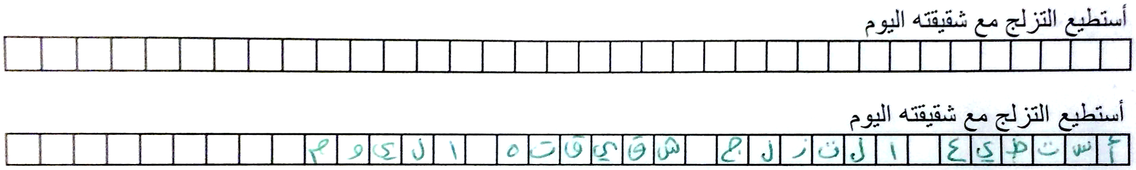
After indicating the lack of research regarding text entry for Arabic Language, a study was conducted on fragmentation text entry for Arabic Language. The main goal was to observe fragmentation text entry and the estimate time taken for that task.

5.2.1 Participants

Fifteen university graduate volunteers, from different Arab countries studying and living in the UK, helped in conducting the survey. Eleven aged 26-35, four aged 36-49, 46.6% females and 73.3% males, and bachelor and M.Sc. degree holders. Their fields of study and work varied greatly. Their main speaking language was Arabic.

5.2.2 Apparatus

Thirty-five Arabic sentences were printed on paper with boxes below, as shown in figures 5.2. The figure also shows the fragmented sentences written by users into those boxes. These sentences were selected from the standard MacKenzie and Soukerof set developed short phrases for the study and translated into Arabic (MacKenzie, I. S. and Soukoreff, R. W., 2003), as shown in appendix D. The study used a paper and a pen, and a stopwatch.



أستطيع التزلج مع شقيقته اليوم

أستطيع التزلج مع شقيقته اليوم

أ س ت ط ي ع ا ل ت ز ل ج ش ق ي ق ت ه ا ل ي و م

Fig 5.2 Fragmentation text entry for Arabic language

5.2.3 Procedure and Results

Participants were asked to fragment words into letters and write them in boxes below the sentence in order to estimate the time taken for that task. Time was recorded from the beginning till the completion of the given sentence. The study showed that participants wrote fragmented letters at an average speed of 10.1 words-per-minute throughout the test session. However, the English handwriting speeds are 20+ words-per-minute, where the average of 31 words-per-minute for memorized text and 22 words-per-minute while copying (Brown, 1988; Bailey, 1996). Consequently, it was indicated that the Arabic handwriting speed is 32-45% the speed of the English handwriting.

5.3 Development and Evaluation of the 12-Key GBAWP

The development of the 12-key GBAWP was to observe gesture text entry and the estimate time taken for that task. The interface appeared like a 12-key physical keypad phone, which used swipe instead of tapping on keys for text entry. It was developed in order to maximize the text area field and to minimize number of the keys displayed on the screen (30 to only 12 keys). Usability test was conducted in which to extract completion times (in seconds) for each character.

5.3.1 Participants

Fifteen volunteer participants, recruited in Egypt by personal contact, took part in conducting the test. Their main speaking language was Arabic. They were Egyptians, aged 20-40, 45% females and 55% males, and bachelor degree holders. Their fields of study and work varied greatly. Only five participants were familiar with touch screens. Also, seven participants were desktop/laptop users.

5.3.2 Apparatus

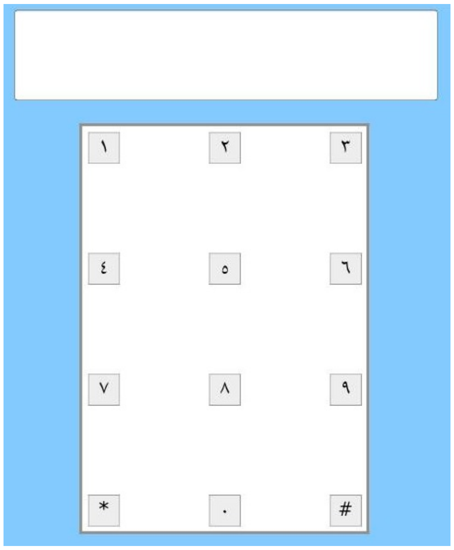


Fig 5.3 12-Key GBAWP

The prototype shown in figure 5.3 was developed in HTML5 and JavaScript. It was displayed on an iPad Mini (2048 x 1536 resolution at 326 pixels-per-inch, 7.9" screen size), running iOS 7.

5.3.3 Procedure

Thirty trials (one successive attempt per trial) were made. Participants were given a random set of Arabic alphabetic letters (one at a time) to type in the prototype using swipe gestures. Swipe gesture is preformed by moving one finger across a touchscreen in a single direction. Each letter had a pre-defined set of swipes, done by connecting two or more virtual keys display on the screen. While they were completing the swipes for the given letter a prediction button appeared with possible predictions of letters to choose from. On completion of swipes, the participant chose from the prediction bar the intended letter. This chosen letter was then added to the text area. This was counted as a successive attempt and he/she was given the next letter to enter afterwards.

5.3.4 Results and Discussion

The results in table 5.1 show that letter ‘ر’ recorded the fastest written letter among all letters. Furthermore, the letter ‘ا’ reported as the second fastest letter, which is the most used letter in the Arabic language. The second, third, and last frequent used letters were ‘ل’, ‘ي’, and ‘ظ’ that recorded 3.3, 4.3, and 7.8 sec, respectively. The slowest letter written was ‘شـ’ and recorded 15.5 sec. Figure 5.4 represents the Arabic letter writing duration in comparison to how frequent the letter appeared in the previously developed Arabic corpus.

Table 5.1 Arabic Letter duration using 12-Key GBAWP

Letter	Time	Freq.	Letter	Time	Freq.	Letter	Time	Freq.	Letter	Time	Freq.
ا	1.60	0.187	ذ	4.72	0.008	ظ	7.85	0.001	ن	7.96	0.0 4
ب	7.26	0.034	ر	.50	.046	عـ	.14	0.019	ه	3.82	0.025
ت	8.04	0.034	ز	2.70	0.011	غـ	7.92	0.011	و	5.62	0.062
ث	0.85	0.011	سـ	6.98	0.026	فـ	7.63	0.019	ي	4.32	0.099
جـ	4.82	0.017	شـ	15.49	0.011	قـ	11.00	0 025	(Space) مسافة	1.58	-
حـ	3.88	0.020	ص	8.86	0.008	كـ	2.70	0.023			
خـ	5.73	0.008	ضـ	10.35	0.006	ل	3.30	0.111	(Delete) حذف	1.48	-
د	2.22	0.028	ط	5.93	0.005	م	6.32	0.056			

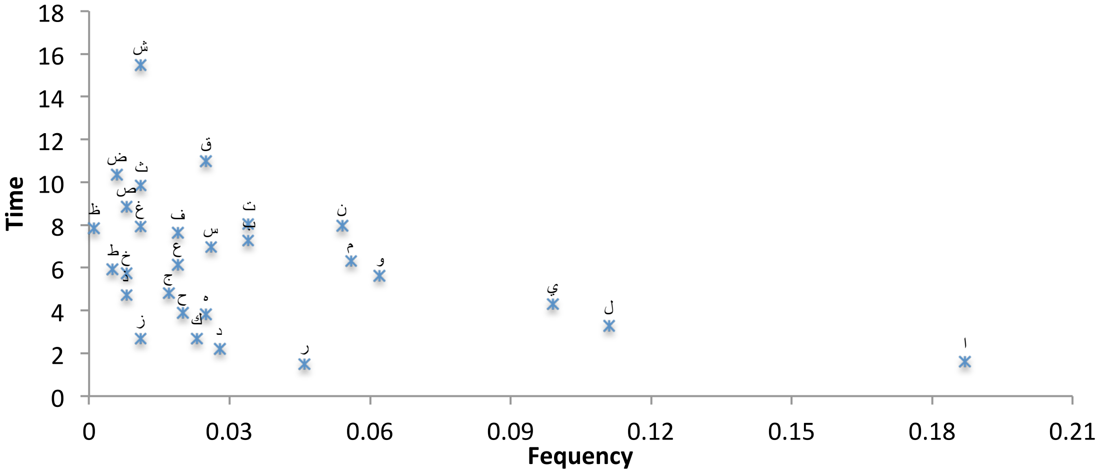


Fig 5.4 Arabic letter writing duration vs. frequency

5.4 Development and Evaluation of the 6-Key GBAWP

In the online recognition field, El-Wakil and Shoukry (1989) studied the structure of Arabic letters and noticed that every Arabic letter has a main stroke and some letters have dots and secondary strokes. However, distinguishing letters from each other is done by the addition of secondary strokes, e.g., dots, in different positions relative to the main stroke as (ـت، ـط، ـظ، ـب). Furthermore, Some Arabic characters contain loops such as (ـف), but no more than two loops may be adjacent share a common link. Considering the Arabic letters characteristics, structure, and maximizing speed, an optimization technique for GBAWP layouts based on dot recognition was used to create a new writing pad capable of fitting in smaller display mobile devices. This technique used swipes to represent the main and secondary strokes, while tapping on virtual keys to add dots to character.

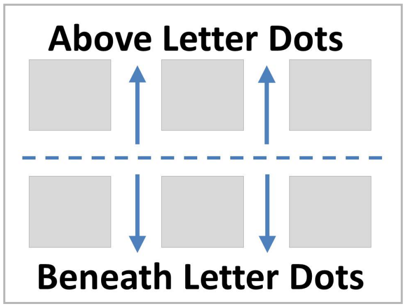


Fig 5.5 Dots Presentation

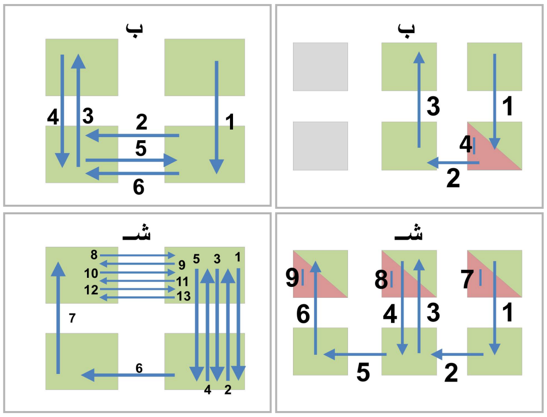


Fig 5.6 Optimization of the “ش” and “ب” letter form

As previously mentioned, some letters (ـب, ـت, ـث, ـجـ, ـي, etc.) have dots above or beneath the main stroke. Accordingly, the interface was divided into two horizontal halves, each representing a different position of dots relative to the main stroke, as shown in figure 5.5. Consequently, this optimization leads to minimizing the number of swipes for letters containing dots in its representation, as shown in figure 5.6. Furthermore, as seen in figure 5.7, letter “ـب” required 6 swipes using 12-Key GBAWP (top left), but 3 swipes and 1 tap using 6-Key GBAWP (top right). Also, letter “ش” required 13 swipes using 12-Key GBAWP (bottom left), still it required 6 swipes and 3 tap using 6-Key GBAWP (bottom right). The blue arrow represented the swipe direction from one virtual key to another, while the short blue line in addition to the red rectangle represented a single tap on a virtual key.

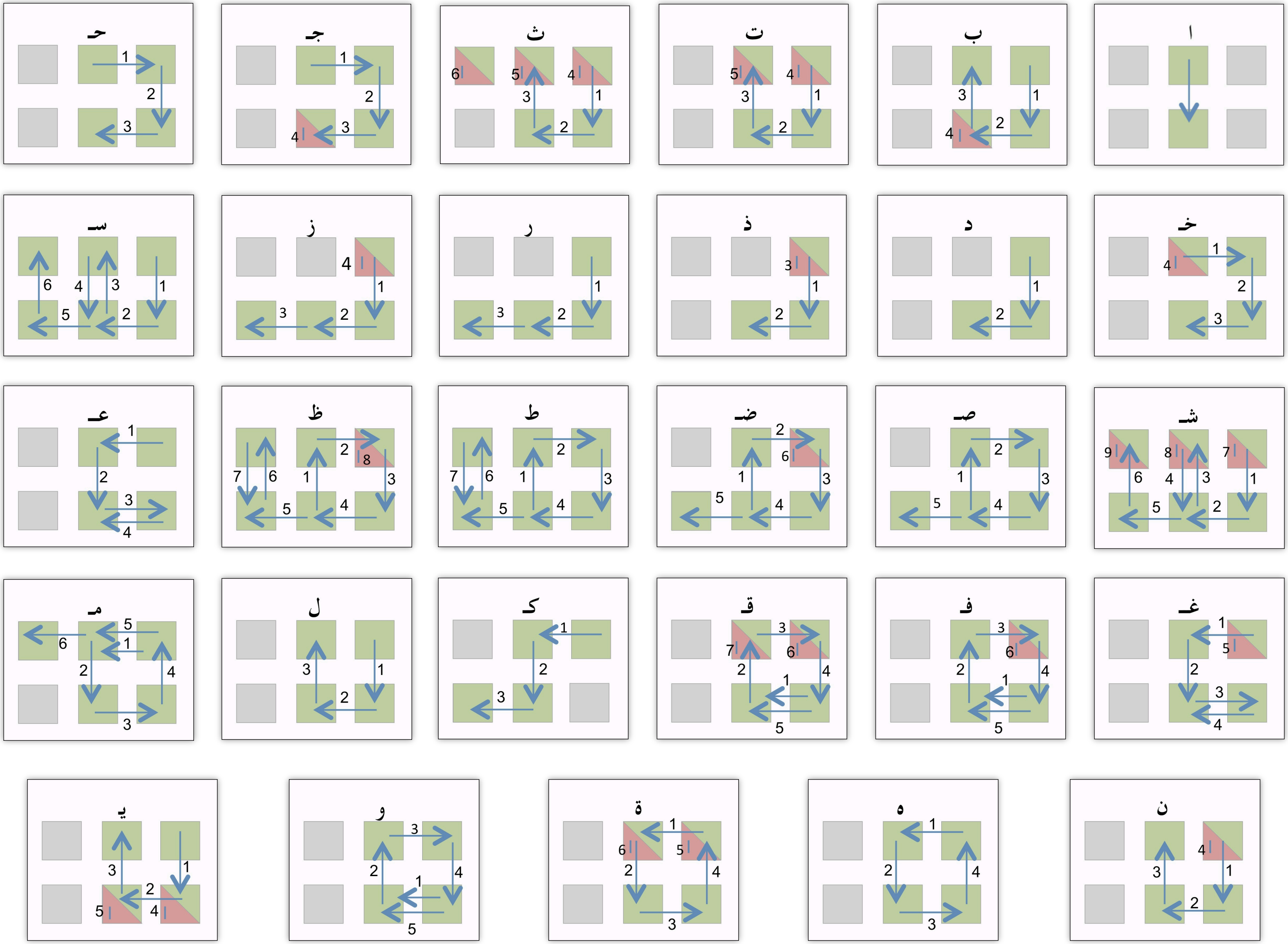


Fig 5.7 GBAWP character form

After modeling the 6-Key and 12-Key GBAWP using the enhanced version of KLM (mentioned in chapter 6), the modeled version of both GBAWPs was done as follows:

1. Calculate the time taken for a swipe/tap.

   - Swipe interaction: $MT_S = 44.78 + 34.64\ ID$, where ID = 2 (12-Key) or 1 (6-Key).
   - Tap interaction: $MT_T = 54.38 + 13.27\ ID$, where ID = 1 (6-Key).

2. Fragment letters to a set of swipes and taps, and accumulate these times to form time-per-letter.

   - Number of Swipe-per-letter: N_S
   - Number of tap-per-letter: N_T
   - Time-per-letter: $T_L = (N_S * MT_S) + (N_T * MT_T)$.

3. Fragment sentences into letters and spaces, calculate number of words, accumulate these times to form time-per-sentence, and calculate words-per-minute.

   - Number of characters: N_C, where each letter or space is counted as one character.
   - Number of words: $N_W = N_C / 5$, where the average word length in Arabic is 5 (Alotaiby et. al, 2009).
   - Time-per-sentence: T_S
   - Words-per-minute: $T_W = T_S / N_W$.

Table 5.2 Arabic Letter duration for modeled 12-Key vs. 6-Key GBAWP

Letter	(6-Key)	(12-Key)	Letter	(6-Key)	(12-Key)	Letter	(6-Key)	(12-Key)
ا	0.79	1.14	ز	3.06	3.42	قـ	5.32	10.27
ب	3.06	6.84	سـ	4.77	7.98	كـ	2.38	3.42
ت	3.74	7.98	شـ	6.79	14.83	ل	2.38	3.42
ث	4.41	10.27	صـ	3.97	7.98	م	4.77	6.84
جـ	3.06	4.56	ضـ	4.65	10.27	ن	3.06	7.98
حـ	2.38	3.42	ط	4.77	5.70	هـ	3.18	4.56
خـ	3.06	5.70	ظ	5.44	7.98	و	3.97	5.70
د	1.59	2.28	عـ	3.18	5.70	ي	3.74	4.56
ذ	2.26	4.56	غـ	3.85	7.98	(Space) مسافة	0.79	1.14
ر	2.38	1.14	فـ	4.65	7.98	(Delete) حذف	0.79	1.14

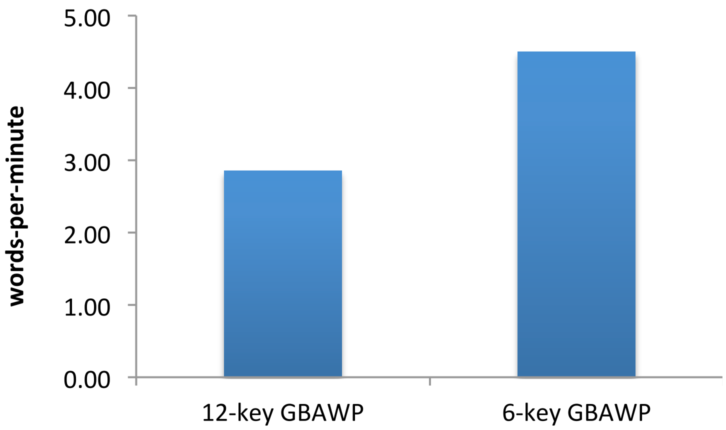


Fig 5.8 Modeled 6-Key vs. 12-Key GBAWP

The results in figure 5.8 show that the estimated words-per-minute are 4.5 and 2.9 on average using 6-Key and 12-Key GBAWP, respectively. Furthermore, the results of table 5.2 show that 11% to 62% less time is taken to type letters using 6-Key GBAWP technique. However, for ‘ر’ it took double the time to write the character.

5.4.1 Implementation of the 6-Key GBAWP on SmartWatch

GBAWP was introduced to enable users to perform Arabic text entry on small display mobile devices. Therefore, the 6-Key GBAWP was implemented on smart watch. The design of prototype slightly changed from the previous 6-Key GBAWP, so as to overcome limitations in the watch's software, as shown in figure 5.9. The display was divided into seven zones. Zones 1 to 6 form large keys while the center zone shows the current input text and acts as a general area for general interactions (letter verification, space, and backspace).

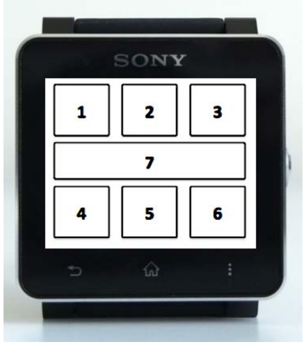


Fig 5.9 6-Key GBAWP zone distribution on the SONY SmartWatch 2

For main text entry the user performed interactions on the watch surface with the input being processed by the text entry system (running on the connected smartphone). Interactions were defined as follows:

- A swipe between two zones from 1 to 6 entered that first zone number and swipe direction (up, down, left, or right).
- A tap on zone 1- 6 entered that zone number.
- A left swipe on the central zone added a space.
- A center tap is used for letter or word completion.
- A right swipe on the central zone is used for backspace.

As previously mentioned, regarding the limitations, any application displayed on the watch was just a mirror of it running on the smart phone. In particular, the watch can neither track a live interaction nor get the length, path, or accurate direction of a gesture. Moreover, the watch only handles basic events and parameters, which are sent back and forth between it and the phone. Consequently, a slight change in the process flow was done in order to overcome these limitations[5]. The process, as shown in figure 5.10, is initiated by handling a touch event occurring on the watch screen. Parameters, such as X/Y position of the finger and direction (up, down, left, or right), are sent back through Bluetooth to the java application running on the phone.

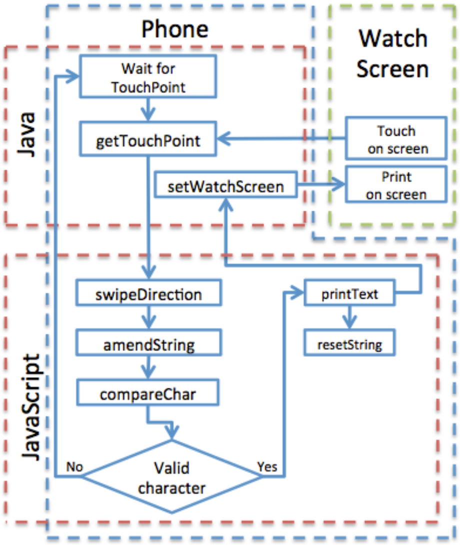


Fig 5.10 6-Key GBAWP Process Flow on the smart watch and the mobile phone

[5] The software was developed in conjunction with Dr Mark Dunlop. Dr Dunlop developed a lightweight Android Java application that displayed buttons on the Sony watch, received user inputs, passed these through to the main JavaScript text entry engine, and finally presented these results back on the watch's screen. The JavaScript code text entry engine was written solely by the candidate.

Afterwards, A function within the developed JavaScript code (running in a webpage window within the java application) receives the data to identify the swipe direction. After detecting the swipe direction, it is amended to a string that is used afterward in the comparison process using the compareChar function. Subsequently, after the conversion of the string set into an Arabic character, the printText function sends the character back to the setWatchScreen function in java application. Finally, the setWatchScreen function updates the watch surface.

5.4.2 Evaluation of the 6-Key GBAWP on SmartWatch

Usability test was conducted in which to extract completion times (in seconds) for each sentence written on the Sony SmartWatch 2 using 6-Key GBAWP.

5.4.2.1 Participants

Fifteen volunteer participants, recruited in Egypt by personal contact, took part in conducting the test. Their main speaking language was Arabic. They were Egyptians, aged 20-35, 40% females and 60% males, and bachelor degree holders. Their fields of study and work varied greatly. Only five participants were familiar with touch screens. Also, nine participants were desktop/laptop users.

5.4.2.2 Apparatus

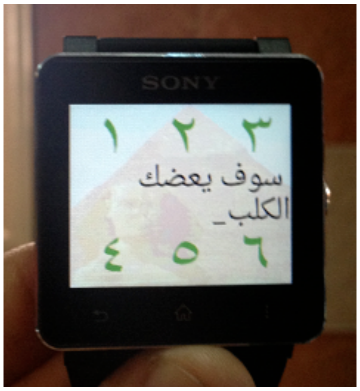


Fig 5.11 6-Key GBAWP on a SONY SmartWatch 2

The prototype shown in figure 5.11 was developed in HTML5, Java and JavaScript. It was displayed on a SONY SmartWatch 2 (220 x 176 pixel resolution, 1.6" screen size), running Android OS 4.0. The watch was connected to a Motorola Moto G (running Android OS 4.4 KitKat) via Bluetooth.

5.4.2.3 Procedure

Participants were given a random set of thirteen Arabic sentences (3 practice sentences + 10 test sentences) to type in the application running on the watch using gesture-based text entry. After conducting the correct set of swipes, letters and afterwards words are written in the text area viewed on the watch's surface. If the user typed in the correct sentence, this was counted as a successive attempt and he/she was given the next sentence to enter.

5.4.2.4 Results and Discussion

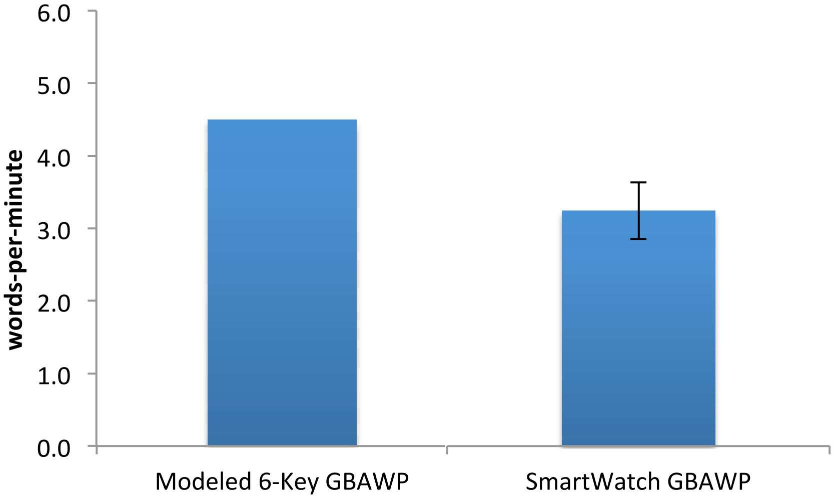


Fig 5.12 Modeled 6-Key vs. SmartWatch GBAWP

The results in figure 5.12 show that the participants executed text entry tasks through the usability test with an average of 3.2 words-per-minute. As mentioned in the previous section, the estimate of 4.5 words-per-minute on average can be achieved on the long term (modeled using KLM). That implies the ability of Arabic language users achieving almost half the hand writing speed (average rate of 10.1 words-per-minute), after practicing, while typing on a 1.6" screen watch.

Moreover, text entry techniques on small screens were compared to handwriting for both English and Arabic Languages. Table 5.3 shows that techniques such as; ZoomBoard, Quikwriting and Three-key text entry achieved from 18-44% of the hand writing speed in the English language, while 6-Key GBAWP achieved 32% of the hand writing speed in the Arabic language.

Table 5.3 Text entry vs. handwriting speed for English and Arabic Language

	English Language		Arabic Language	
Text entry on small screens	ZoomBoard	7.6 wpm (Oney et al., 2013)	6-Key GBAWP	3.2
	Quikwriting	4 wpm (Isokoski and Raisamo, 2004)		
	Three-key text entry	9.6 wpm (MacKenzie, 2002)		
Handwriting	22 wpm while copying (Brown, 1988)		10.1 wpm	

5.5 Summary

This chapter illustrates Gesture-based text entry as a method used for mobile devices. Its success and acceptance is critically dependent on the reliability of gesture recognition. The gesture recognition of the GBAWP is accomplished through a sequence of touched points or swipes on the screen. In order to maximize the text area field and minimize the number of keys displayed on the screen, a 12-key GBAWP interface was introduced appearing like a 12-key physical keypad phone.

Moreover, considering the Arabic letters characteristics, structure, and maximizing speed, a 6-key GBAWP layout based on dot recognition was introduced. After conducting usability tests on both the 12-key and 6-key GBAWP, it was found that users could perform text entry on mobile devices using the 12-key GBAWP with an estimate of 2.9 words-per-minute on average. Furthermore, they also executed text entry tasks on a Sony SmartWatch 2 with an average of 3.2 words-per-minute, with an estimation of 4.5 words-per-minute, on average, on the long term. While entry speeds were slow, users found it easy to use.

Therefore, Gesture-based technique enables users to perform Arabic text entry on small display mobile devices and watches, which is done using both the 12-key and 6-key GBAWP. The next chapter will introduce an enhancement on KLM model to handle interactions on mobile devices and tablets.

6. Enhancing KLM on Mobile Devices

After presenting an enhancement of form navigation using Panning and Zooming on mobile devices, an optimized Arabic keyboard layout, and the 12-key and 6-key GBAWP. A quantitative prediction model must be introduced to evaluate these techniques and keyboards. Models, such as GOMS (Goals, Operators, Methods, and Selection rules) and KLM (Keystroke-Level Model) have been shown to be useful tools in modeling interaction and in deciding between, and filtering out designs (Card et. al, 1983). KLM is used to estimate the time taken to complete simple data input tasks by combining small timing constants. On physical-key devices it has been widely used to predict task times (e.g. text entry for Korean Language (Myung, 2004), Text Entry Speed on 12-button Phone Keypads (Pavlovych and Stuerzlinger, 2004) etc.).

Many enhancements to KLM have been proposed in the literature to be able to evaluate different techniques. However, there has been little published research that improves user behavior modeling techniques that are capable of estimating the time taken to achieve common interactions performed such as swipe, zoom, or tapping, on mobile touchscreen devices. Tran et al. (2013) conducted a study of pinch and spread gestures on a phone and a tablet device. They used Fitts' Law in order to characterize execution time for pinch and spread gestures on mobile devices (57% of the original data was used in the analysis).

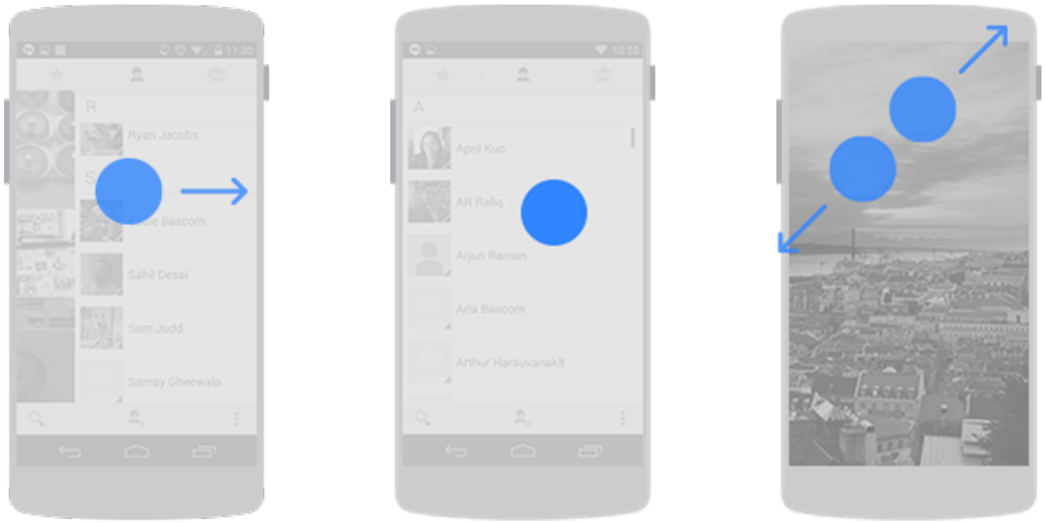
The proposed work not only focuses on enhancing KLM by extending it with three operators, but also compares the results with the previous work of Tran et al. (2013). An investigation is done on modeling these interactions performed on mobile devices and tablets. The proposed model is based on suggested times derived from Fitts' Law modeling and analysis of how well these interactions fit Fitts' Law. It also gives an enhancement for developers of smart-phone and tablet applications to predict user interaction times without even needing to create prototypes.

In this chapter, the work done on enhancing KLM to handle interactions on mobile devices and tablets will be shown. The first subsection will cover the concept and methodology of building the Swipe, Tap, and Zoom prototypes. The second subsection will focus on developing, testing, and outcomes of the prototypes. The final subsection will illustrate a brief summery.

6.1 Concept and Methodology of Prototypes

The development of three prototypes came in the process of enhancing KLM derived from GOMS model. These prototypes reflected three main interactions, as shown in figure 6.1, used nowadays on tablets and mobile devices;

1. Swipe interaction, which enables the user to do different swipe gestures (short or long) to scroll overflowing content, or navigate between views (e.g. scrolling through a document/websites etc.).
2. Tap interaction, which estimates the user's tapping response time in comparison to his/her finger travel distance from a home position (e.g. to estimate typing speed or pointing to a target etc.).
3. Zoom interaction, which is used to cause text or other graphics in a window or frame to appear larger on the screen (e.g. zooming, panning a map/photo etc.).



Swipe action: press, move, lift.

Tap action: press, lift.

Zoom action: 2-finger press, move outwards, lift.

Fig 6.1 Three main interactions to enhance KLM on mobile devices

The three prototypes[6] were built to estimate the time taken to conduct these interactions. Fitts' Law (Paul Fitts, 1954) as a model is used to analyze the general case of estimated time taken for these interactions. It is primarily used in human–computer interaction and ergonomics that predicts the time

[6] The study systems were developed in HTML and JavaScript by the candidate.

required to rapidly move to a target area, it is a function of the distance to the target and the size of the target. Of various formulations, the Shannon formulation is one of the mathematical formulas of Fitts's law, which was proposed by Scott MacKenzie (1992). It was named for its resemblance to the Shannon–Hartley theorem:

$$T = a + b * ID, \text{ where ID} = \log_2\left(\frac{\mathrm{D}}{\mathrm{W}} + 1\right).$$

- T is the average time taken to complete the movement. It may also be referred to as MT (mean movement time).
- a represents the incorporating reaction time and/or the time required to execute the operator.
- b stands for the inherent speed of the device.
- D is the distance from the starting point to the center of the target.
- W is the width of the target measured along the axis of motion.

The constant "1" in the formula made a difference from Fitts's original form, especially for low values of the ratio D/W, given the advantage that the ID (measured in bits) is always non-negative. Bi et al. (2013) introduced the FFitts model — an extension of Fitts' law to accurately model finger touch input for small target acquisition tasks. However, this work investigated the ability to model the finger touch input for different target sizes and movement distances. An introduction on Fitts' Law in addition to a review on related studies to it in the field of HCI will be illustrated in appendix A.

6.2 Development and Evaluation of Prototypes

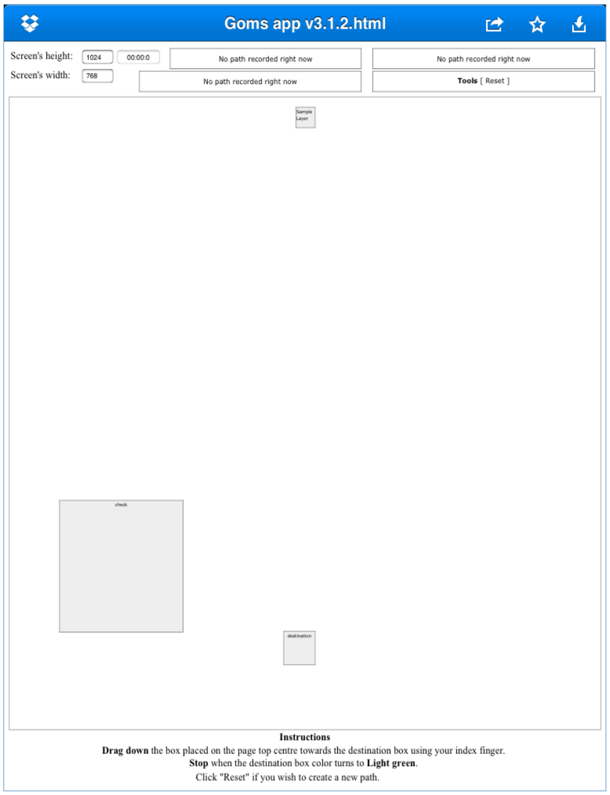
Three usability tests were conducted in which to extract completion times (in seconds), for each interaction, in terms of different target size and distance from the starting point to the center of the target (in pixels).

6.2.1 Participants

Thirty volunteer participants, recruited in Egypt by personal contact, took part in conducting the Swipe and Zoom usability tests, while twenty-one volunteer participants conducted the Tap usability test. Participants were Egyptians, aged 22-39, 35% females and 65% males, and all are bachelor degree holders. Their fields of study and work varied greatly. Eighteen participants were familiar with touch screens. Also, twenty-two participants were desktop/laptop users.

6.2.2 Apparatus

The prototypes shown in figure 6.2 were developed in HTML5 and JavaScript. It was displayed on an iPad Mini (2048 x 1536 resolution at 326 pixels-per-inch, 7.9" screen size), running iOS 7.

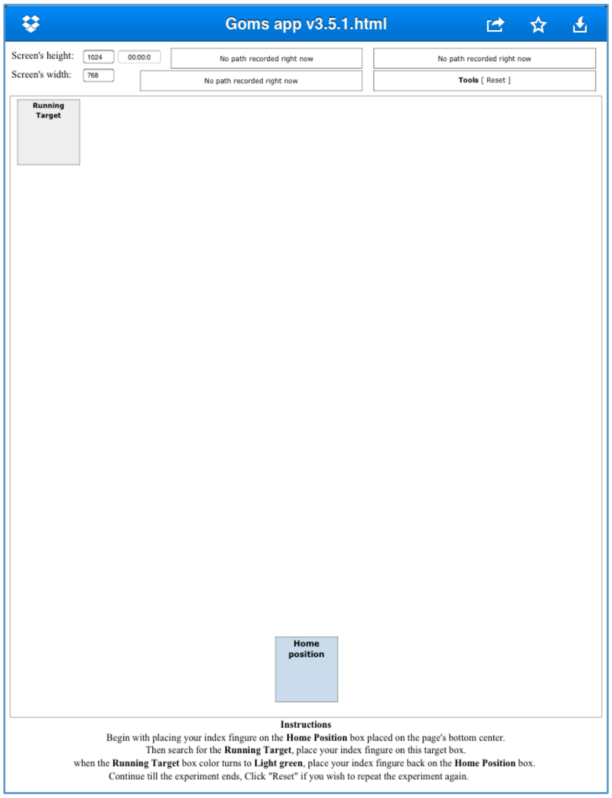


Screen's height: 1024 00:00:0 No path recorded right now No path recorded right now

Screen's width: 768 No path recorded right now **Tools** [Reset]

Instructions

Drag down the box placed on the page top centre towards the destination box using your index finger.
Stop when the destination box color turns to **Light green**.
Click "Reset" if you wish to create a new path.

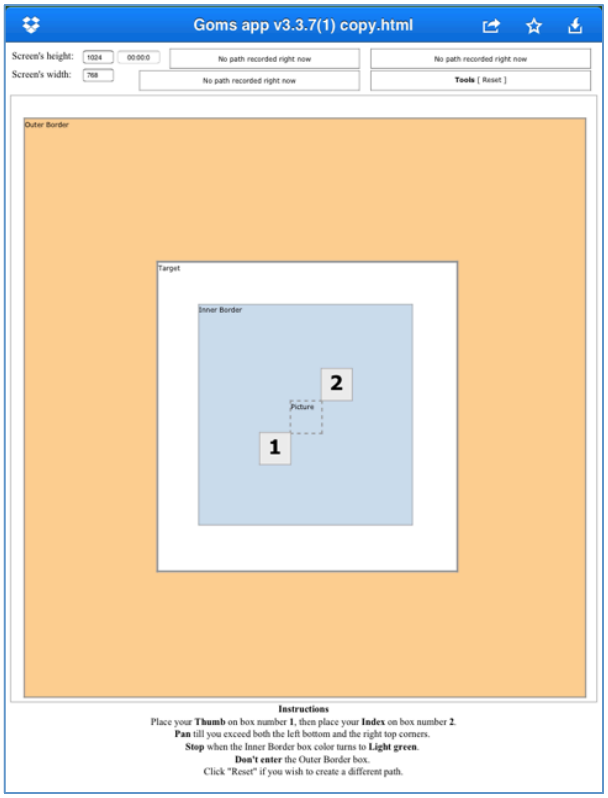


Screen's height: 1024 00:00:0 No path recorded right now No path recorded right now

Screen's width: 768 No path recorded right now **Tools** [Reset]

Instructions

Begin with placing your index fingure on the **Home Position** box placed on the page's bottom center.
Then search for the **Running Target**, place your index fingure on this target box.
when the **Running Target** box color turns to **Light green**, place your index fingure back on the **Home Position** box.
Continue till the experiment ends, Click "Reset" if you wish to repeat the experiment again.



Screen's height: 1024 00:00:0
Screen's width: 768

No path recorded right now

No path recorded right now

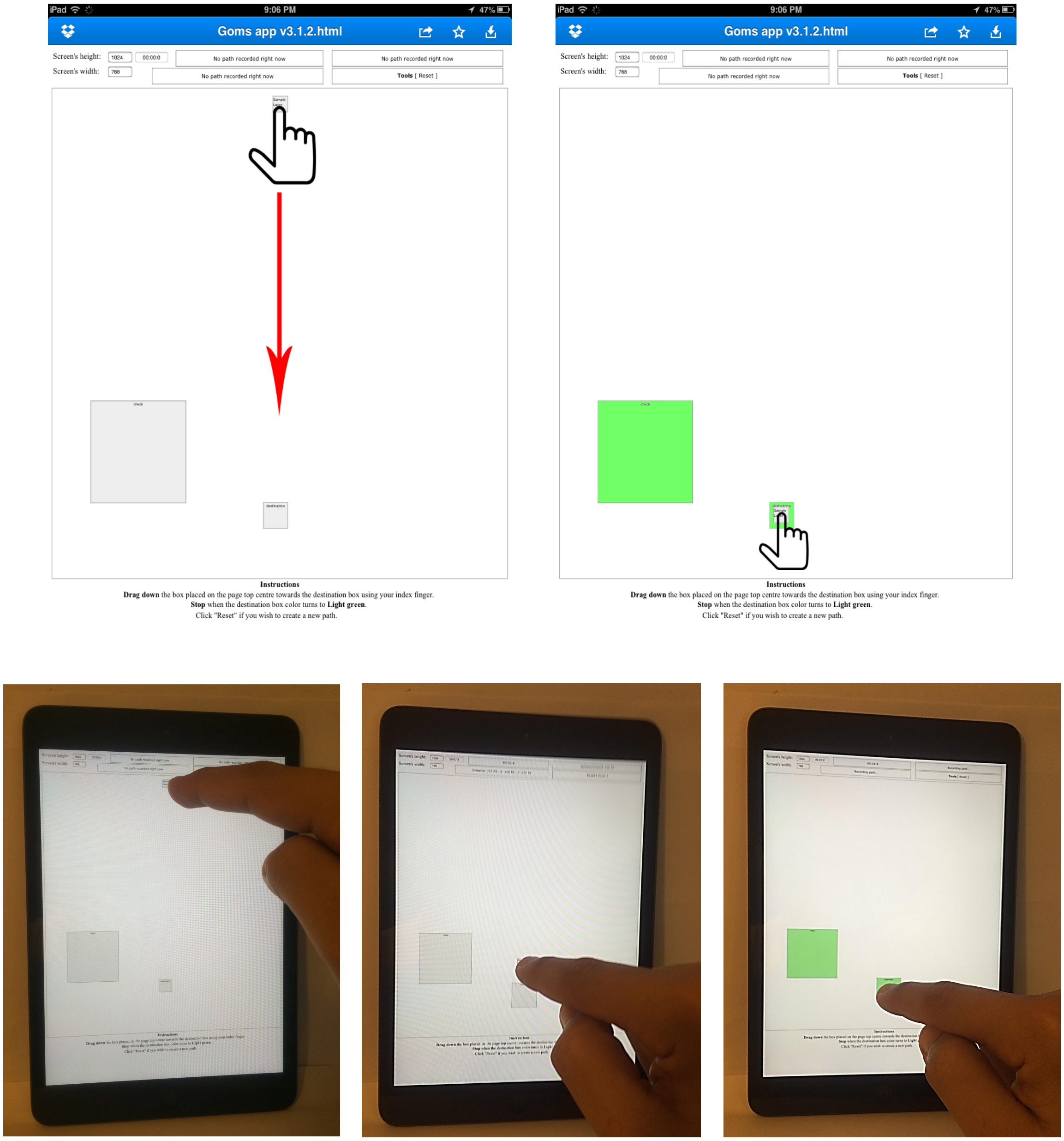
No path recorded right now

Instructions
Place your **Thumb** on box number **1**, then place your **Index** on box number **2**.
Pan till you exceed both the left bottom and the right top corners.
Stop when the Inner Border box color turns to **Light green**.
Don't enter the Outer Border box.
Click "Reset" if you wish to create a different path.

Fig 6.2 Swipe (top-left), Tap (top-right), and Zoom (Bottom) Prototypes

6.2.3 Procedure

The Swipe study target was to observe swipe time in seconds and distance in pixels. Three trials (sixteen successive attempts per trial) were made. As shown in figure 6.3, users dragged down a box placed on the top center towards the destination box placed on the bottom center of a web page using the participant’s index finger. The participant stopped and removed his/her index finger from the screen when the destination box color turned to light green. This was recorded as a successful attempt.



Screen's height: 1024 00:00:0 No path recorded right now No path recorded right now
Screen's width: 768 No path recorded right now **Tools** [Reset]

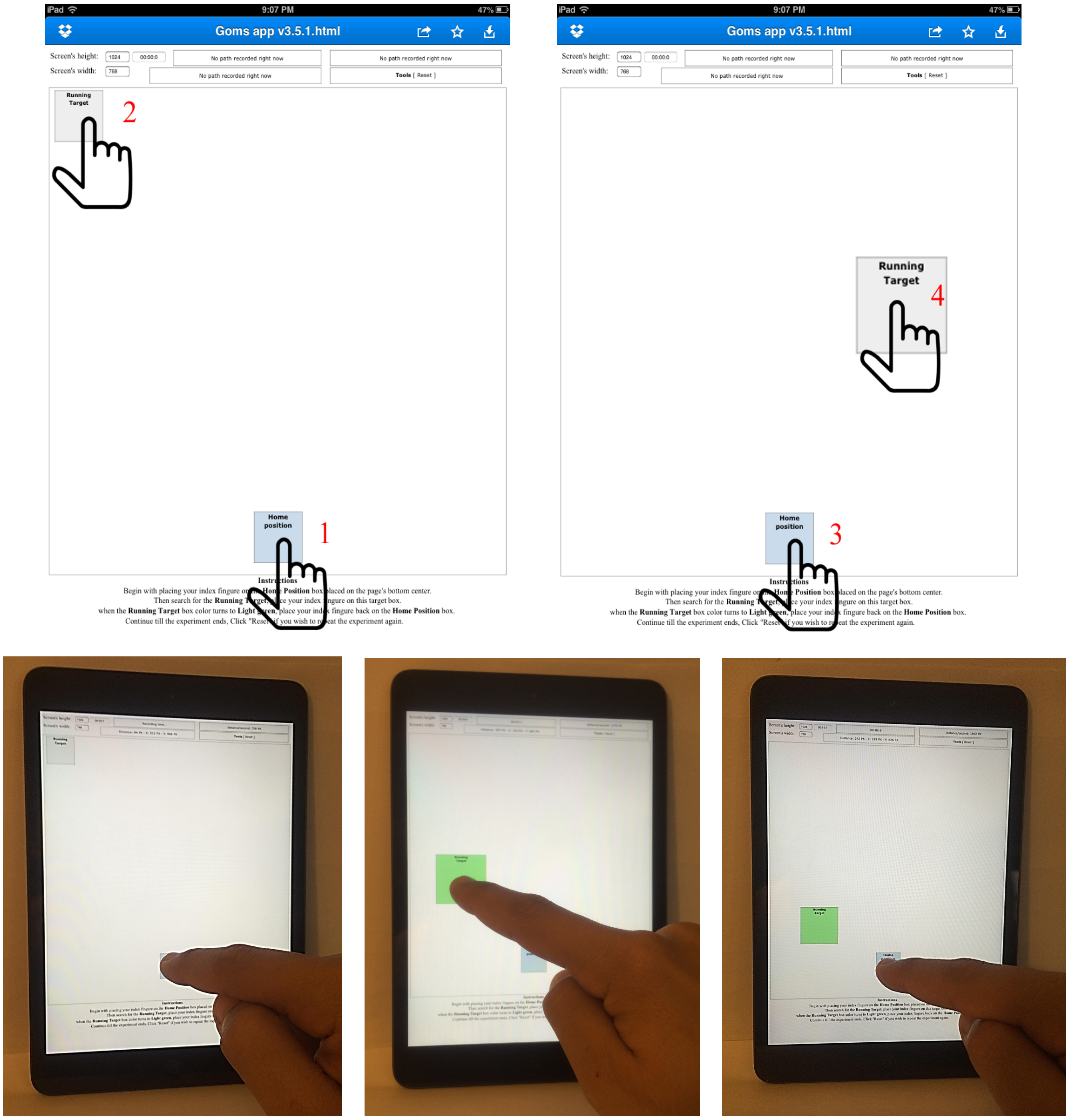
Instructions
Drag down the box placed on the page top centre towards the destination box using your index finger.
Stop when the destination box color turns to **Light green**.
Click "Reset" if you wish to create a new path.

Screen's height: 1024 00:00:0 No path recorded right now No path recorded right now
Screen's width: 768 No path recorded right now **Tools** [Reset]

Instructions
Drag down the box placed on the page top centre towards the destination box using your index finger.
Stop when the destination box color turns to **Light green**.
Click "Reset" if you wish to create a new path.

Fig 6.3 Swipe usability test scenario

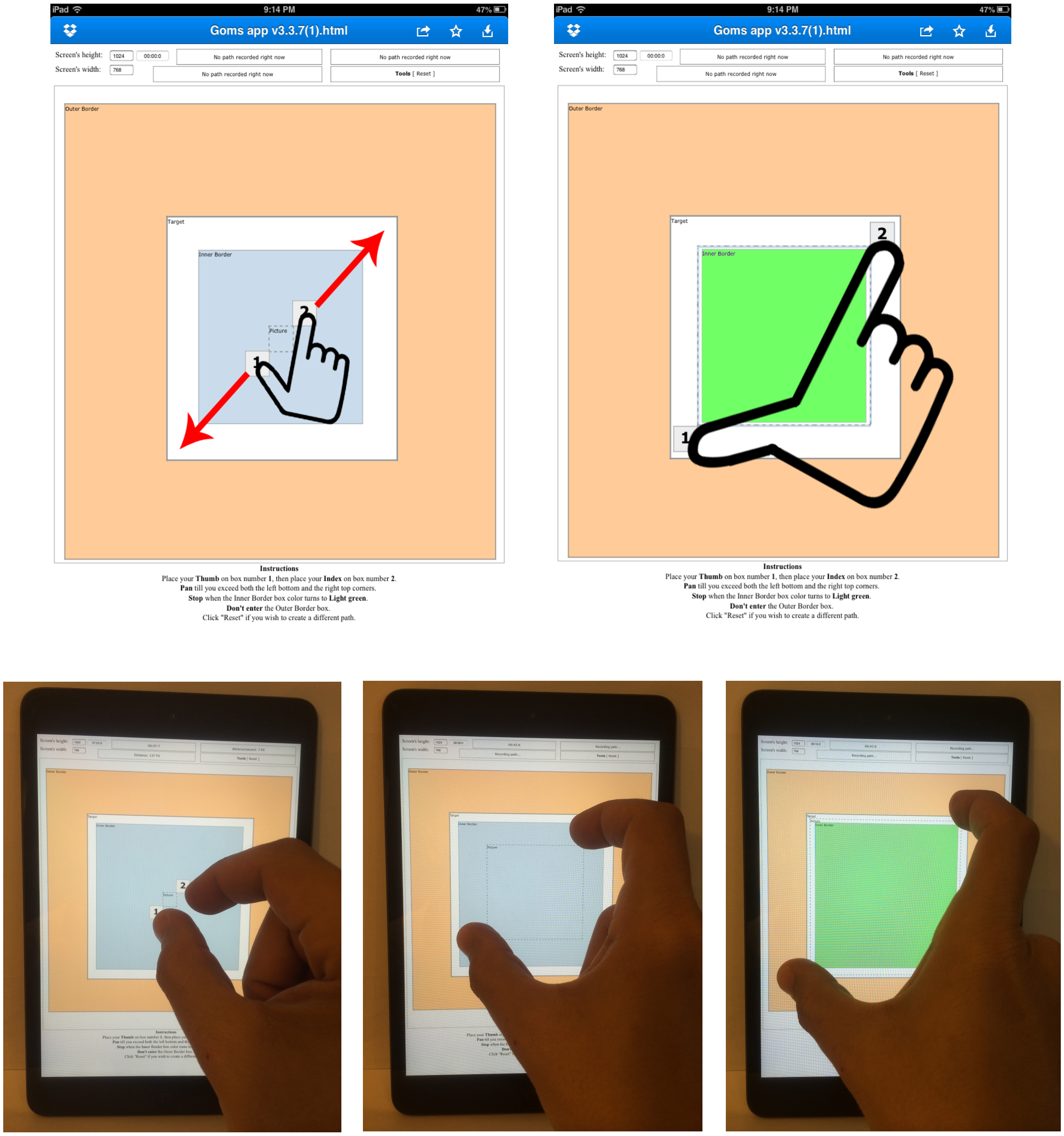
The Tap study was to record the participant's track and touch time in seconds, target size and distance in pixels. Two trials (five successive attempts per trial) were made. As shown in figure 6.4, each by placing the index finger of the participant on the "Home Position" box placed on the bottom of the page. The participant searched for the "Running Target" box and placed his/her index finger on this target box. He/she removed his/her index finger from the "Running Target" box when its color turned to Light green. Afterwards, the participant placed his/her index finger back on the "Home Position" box. This was recorded as a successful attempt.



Begin with placing your index fingure on the Home Position box placed on the page's bottom center.
Then search for the **Running Target**, place your index fingure on this target box.
when the **Running Target** box color turns to **Light green**, place your index fingure back on the **Home Position** box.
Continue till the experiment ends, Click "Reset" if you wish to repeat the experiment again.

Fig 6.4 Tap usability test scenario

The Zoom study targeted at recording the participant's zoom time in seconds, target size and distance in pixels. One trial (twenty six successive attempts per trial) was made. As shown in figure 6.5, each by placing the thumb finger of the participant on the box number 1 and his/her index finger on the box number 2. The participant zoomed till he/she exceeded both the left bottom and the right top corners of the border box, within the target box. He/she stopped and removed his/her thumb and index fingers from the screen when the border box color turned to Light green. This was recorded as a successful attempt.



Screen's height: 1024 00:00:0

Screen's width: 768

No path recorded right now

No path recorded right now

No path recorded right now

Tools [Reset]

Instructions

Place your **Thumb** on box number **1**, then place your **Index** on box number **2**.

Pan till you exceed both the left bottom and the right top corners.

Stop when the Inner Border box color turns to **Light green**.

Don't enter the Outer Border box.

Click "Reset" if you wish to create a different path.

Instructions

Place your **Thumb** on box number **1**, then place your **Index** on box number **2**.

Pan till you exceed both the left bottom and the right top corners.

Stop when the Inner Border box color turns to **Light green**.

Don't enter the Outer Border box.

Click "Reset" if you wish to create a different path.

Fig 6.5 Zoom usability test scenario

6.2.4 Results and Discussion

6.2.4.1 The Swipe usability study

As shown in table 6.1 and figure 6.6, target width and target amplitude varied across four levels resulting in IDs of 1 to 4 bits. The target width varied between 50, 100, 200, and 400 pixels, while target amplitude varied between 100, 200, 400, and 800 pixels. Mean movement time ranged from 0.2 sec to 16.1 sec with each score derived from over 1440 observations. Correlating MT with ID yields $R^2 = 0.911$ ($p<0.1$). Regressing MT on ID yields the following prediction equation for movement time (ms):

$$MT = 9.46 + 55.83\ ID.$$

Table 6.1 target widths and amplitudes for each ID in Swipe study

IDs	1	2	3	4
Target widths (in pixels)	50, 100, 200, & 400	50, 100, 200, & 400	50 & 100	50
Target amplitudes (in pixels)	100, 200, & 400	200, 400, & 800	400 & 800	800

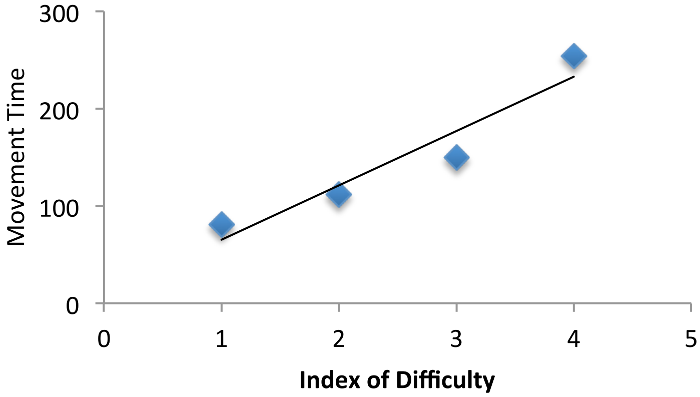


Fig 6.6 Movement Time vs. Index of difficulty for Swipe usability test

6.2.4.2 *The Tap usability study*

As shown in table 4.2 and figure 4.7, the target width varied from 50 to 250 pixels, while target amplitude varied from 125 to 900 pixels, to form four levels resulting in IDs of 1 to 4 bits. Mean movement time ranged from 0.5 sec to 4.7 sec with each score derived from over 210 observations. Correlating MT with ID yields $R^2 = 0.985$ ($p<0.05$). Regressing MT on ID yields the following prediction equation for movement time (ms):

$$MT = 52.12 + 14.62\ ID.$$

Table 6.2 target widths and amplitudes for each ID in Tap study

IDs	1	2	3	4
Target widths (in pixels)	150 - 250	50 - 250	75 & 100	50
Target amplitudes (in pixels)	125 - 400	200 - 900	700 & 900	700

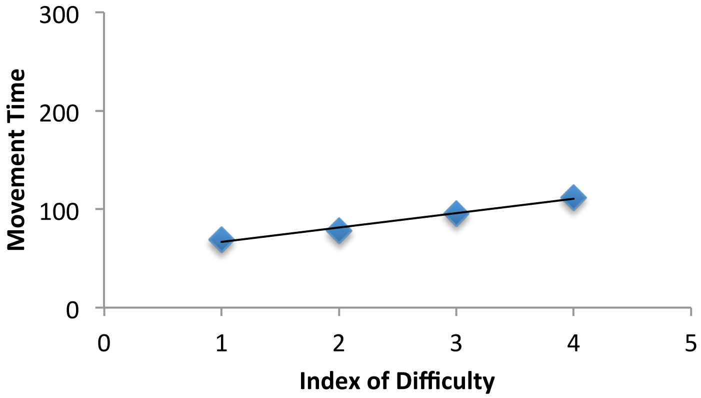


Fig 6.7 Movement Time vs. Index of difficulty for Tap usability test

6.2.4.3 The Zoom usability study

As shown in table 6.3 and figure 6.8, target width and target amplitude varied across four levels resulting in IDs of 1 to 4 bits. The target width varied between 50, 100, and 150 pixels, while target amplitude varied from 100 to 550 pixels with fixed step of 50 pixels. Mean movement time ranged from 0.6 sec to 12.3 sec with each score derived from over 1440 observations. Correlating MT with ID yields $R^2 = 0.971$ ($p<0.05$). By comparing the results with the previous work of Tran et al. (2013), both outcomes are largely in line with slightly different prediction equation and higher correlating MT with ID (R^2 Zoom > 0.884). Regressing MT on ID yields the following prediction equation for movement time (ms):

$$MT = 114.86\ ID - 20.45.$$

Table 6.3 target widths and amplitudes for each ID in Zoom study

IDs	1	2	3	4
Target widths (in pixels)	100 & 150	50, 100, & 150	50 & 100	50
Target amplitudes (in pixels)	100, 150, & 200	100, 150, 200, 250, 300, 350, 400, & 450	250, 300, 350, 400, 450, & 500	550 & 500

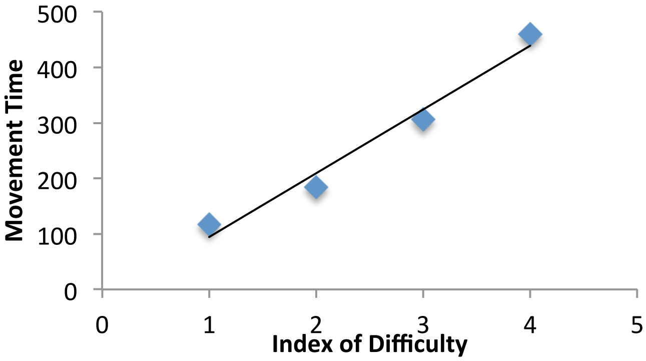


Fig 6.8 Movement Time vs. Index of difficulty for Zoom usability test

6.2.4.4 Defect in the 4th Index of Difficulty Level on small screens

It was noticed in the Swipe and Zoom usability tests that the 4th ID level does not fit in the straight line, which passes by the other 3 levels of ID. Moreover, the constant "a", representing the incorporating reaction time and/or the time required executing the operator, in the prediction equation for movement time of the Zoom usability test is negative. However, by excluding the 4th ID level, as shown in figure 6.9, regressing MT on ID yields the following prediction equation for movement time (ms):

- Swipe interaction: MT = 44.78 + 34.64 ID.
- Tap interaction: MT = 54.38 + 13.27 ID.
- Zoom interaction: MT = 13.75 + 94.33 ID.

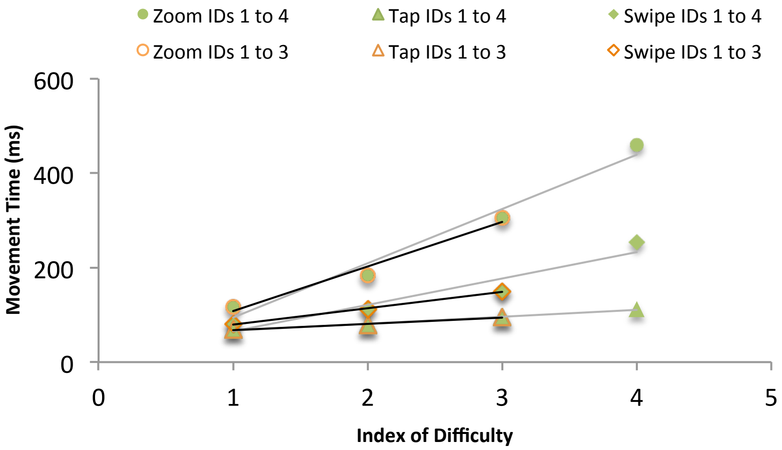


Fig 6.9 MT vs. ID for Swipe, Tap, and Zoom tests with IDs of 1 to 3 and 1 to 4 bits.

Table 6.4 R^2 values for interaction execution times

Interaction	R^2
Swipe usability test with IDs of 1 to 3	0.995 ($p<0.05$)
Swipe usability test with IDs of 1 to 4	0.911 ($p<0.1$)
Tap usability test with IDs of 1 to 3	0.972 ($p<0.05$)
Tap usability test with IDs of 1 to 4	0.985 ($p<0.05$)
Zoom usability test with IDs of 1 to 3	0.972 ($p<0.05$)
Zoom usability test with IDs of 1 to 4	0.971 ($p<0.05$)

The correlating MT with ID for the Swipe usability test yields $R^2 = 0.995$ ($p<0.01$) instead of $R^2 = 0.911$ ($p<0.1$) as shown in table 6.4, which gives a better-fit and more precise prediction equation. Further investigation may also reveal other defects in fitting the ID's greater than 4 on small screens. As shown the poor fit of the 4th ID level is indicative that increasing in the spread distance required for a Zoom task of this size is too large for comfort, where the target amplitude is between 500px and 550px (at about 98 mm) while the target width is 50px.

This result parallels what Hoggan et al. (2013) found for pinch, where large pinch distance of over 90mm took significantly longer time to complete tasks and produced an increase in ergonomic failure rate. Below 90mm was suggested as an optimal maximum extension distance for a dual-touch operation. Those both results imply taking into consideration the ranges of thumb and index fingers extension in any one handed pinch manipulation.

6.3 Summary

This chapter introduces an enhancement to KLM (Keystroke-Level Model), a quantitative prediction model predicting the user's behaviour in low-level tasks. This was acomplished by extending it with three new operators describing interactions on mobile touchscreen devices and tablets. The approach is based on Fitts' Law to identify a performance measure estimate equation for each of the introduced interactions. Three prototypes were developed to serve as a test environment in validating Fitts equations and estimating the parameters for these interactions. Three-thousand and ninty observations took place with a total of 51 users. Based on results, the following approximate movement time for KLM are suggested as shown; a short untargeted swipe will take approximately

0.7 sec, a half-screen sized zoom will take approximately 2.0 sec, and an icon pointing from a home position will take approximately 0.8 sec.

The studies confirmed that most interactions fitted well with Fitts' Law. On the other hand, it was noticed that a poor-fit for some high ID operations indicating a possible maximum comfort limit for these tasks. However, this enhancement in KLM, as a quantitative prediction model, extended it with new operators necessary to describe interactions used on mobile devices and tablets. In the next chapter a final conclusion is given regarding the proposed work.

7. Conclusion and Future Work

Human–computer Interaction (HCI) researchers always try to enhance the interaction between users and computers by introducing new techniques to accomplish tasks in an easy, fast, and accurate way. This improvement must involve understanding the mentality, ability, and need of users, in order to find an efficient way of interaction. In this research, counters of the EAC were the inspiration behind seeking enhancement in interaction held on mobile devices into two key areas of form completion and improved Arabic text entry.

The objective of this research is to apply improvement in techniques that could lead to the achievement of tasks in an easy, fast, and accurate way. Four improvements were taken into consideration in sequence: (1) minimizing large forms to fit small mobile device screens and easing form navigation process, (2) optimizing Arabic keyboard layout to suit Arabic language users, (3) introducing Gesture-based Arabic writing pads that fit small mobile device screens and smart watch surfaces, and (4) enhancing quantitative prediction models to overcome the defect in modeling interactions on mobile devices. These improvements were implemented in a way that make the users capable of increasing the rate of achieving tasks despite their lack of experience dealing with mobile devices and tablets.

The research shows an improvement of form navigation on mobile devices. The approach is based on computerizing forms and using Panning and Zooming as a navigation technique. However, there were considerable challenges in reducing the size of the paper forms to fit mobile devices and introducing fast navigation technique. The Arabic Language users' mentality, ability, and need were the inspiring factors behind the introduction and improvement in different fields; data entry, quantitative prediction model, and form fill-in, in order to reach the targeted objectives.

Furthermore, this research presents a new design of an Arabic keyboard layout for effective text entry on touch screen mobile phones. The approach is based on Pareto front optimization method in the process of optimizing layout. Three metrics were taken into consideration in parallel: (a) speed of text entry, (b) error correction tap interpretation clarity, and (c) familiarity to the vertically aligned alphabetically ordered layout. Moreover, a 12-key GBAWP interface was introduced appearing like a

12-key physical keypad phone, in order to maximize the text area field and minimize the number of keys displayed on the screen. In respect to the Arabic letters characteristics, structure, and maximizing speed, a 6-key GBAWP layout based on dot recognition was introduced.

Finally, this work introduces an enhancement to KLM, a quantitative prediction model predicting the user's behaviour in low-level tasks. This was acomplished by extending it with three new operators describing interactions on mobile touchscreen devices and tablets. The approach is based on Fitts' Law to identify a performance measure estimate equation for each of the introduced interactions.

In this chapter, final conclusion about the accomplished work will be illustrated. This chapter is divided into three sections. The first contains the summary of the proposed research results. The second section embodies the future work recommended to improve the used techniques. The final section contains the key contributions and publications.

7.1 Summary

The proposed work showed an improvement of form navigation on mobile devices. It was concluded that using Panning and Zooming technique scored 31 minutes on average completion tasks time. In comparison to the tabbed navigation technique, it was 4 minutes less. Eventually, using the Panning and Zooming technique showed indicatively lower workload, which reinforces its superiority on using tabbed navigation technique in form navigation.

Moreover, it focused on introducing a new optimized Arabic mobile keyboard layout, using Pareto Front optimization method in the process of optimizing layout. Improvement in typing speed was noticed and measured in formal test sessions with Arabic Language users compared to the Apple Mac Arabic Keyboard layout. In addition to this initial benefit, the design of the keyboard to support faster and more accurate entry should lead to even higher medium and long-term speed. Users' preferences were obtained in favor of the optimized layout, as key distribution is more familiar to the vertically aligned alphabetically ordered layout. Furthermore, comparing Arabic Language users with typical Western users, having a wider variation in standard keyboard layouts, some users still suggested that it would take more time practicing to gain more familiarity with key locations.

Additionally, after conducting usability tests on both the 12-key and 6-key GBAWP, it was found that Arabic Language users could perform text entry on mobile device using the 12-key GBAWP with an estimate of 2.9 words-per-minute on average. Furthermore, they also executed text entry tasks on a Sony SmartWatch 2 with an average of 3.2 words-per-minute. This could increase to an estimate of 4.5 words-per-minute on average, on the long term. While entry speeds were slow, users found it easy to use and it supports largely eyes free interaction. Therefore, Gesture-based technique enables users to perform Arabic text entry on small display mobile devices and watches using both the 12-key and 6-key GBAWP.

Finally, after conducting three usability tests, which estimated the movement time for certain interactions, were done using Fitts' Law. Furthermore, three equations were formed to estimate the movement time for three interactions; Swipe, Tap, and Zoom. The ID was the variable used to be submitted in any of the three equations in order to calculate the movement time. A total of 3090 observations took place with a total of 51 users. Based on results, the following approximate movement times for KLM were suggested. A short untargeted swipe will take approximately 0.7 sec, a half-screen sized zoom will take approximately 2.0 sec, and an icon pointing from a home position will take approximately 0.8 sec. Poor-fit was found for some high ID operations indicating a possible maximum comfort limit for these tasks. However, this enhancement in KLM, as a quantitative prediction model, extended it with new operators necessary to describe interactions used on mobile devices and tablets.

7.2 Future work

Many improvements can be introduced in different fields; data entry, quantitative prediction models and form fill-in, in order to enable the user to increase the rate of achieving tasks. Some of them are:

- Continue with the usability tests of the Optimized Arabic Keyboard to make sure that it could totally overcome the default Arabic keyboard.
- Longer field-based trials are needed to confirm the results of 12-key and 6-key GBAWPs and to assess fully that they can reach their estimated time for words-per-minute.

- Optimize 6-key GBAWP to reach higher speed by minimizing number of swipes per letter (a prototype was developed but not tested on the Sony smartwatch 2, due to watch's programing limitations).
- Enhance KLM by extending it with more new operators (physical movement etc.) necessary to describe other interactions used on mobile devices and tablets.
- Further investigation to reveal other defects in Fitts' Law fitting the ID's greater than 4 on small screens.
- Conduct more experiments with different devices of different screen dimensions and/or touch sensitivity to investigate possible differentiations to the reported results regarding KLM enhancement.
- Longer field-based trials are needed to confirm the results of form navigation and to assess fully whether a Panning and Zooming based tablet application could replace paper-based forms.
- Raise EAC counters awareness to achieve form completion, to give the user the ability to know which parts in the form are completed and which are still not filled in.
- Attempt to decrease the working memory in the presence of context awareness and Panning and Zooming navigation technique used in EAC Form Fill-in.

7.3 Key Contributions and Publications

In conclusion, the key contributions of this research were:

- A new semi-optimized keyboard layout for Arabic language touchscreen mobile devices that is Arabic Language user focused and tested (El Batran & Dunlop, 2013) [Appendix B.1].
- A study of working practices of counters conducting the EAC, a comparison of form navigation techniques (Tabbing vs. Panning and Zooming), and an improvement in form fill-in behavior of computerized large forms displayed on mobile devices and tablets (El Batran & Dunlop, 2014) [Appendix B.2].
- An enhancement in KLM using Fitts' law to model three touchscreen actions on mobile devices and tablets, which includes an explanation of spread distance for a dual-touch manipulation El (El Batran & Dunlop, 2014) [Appendix B.3].
- A watch surface Gesture-based text entry method for Arabic Language users, which is suitable for the use on small display mobile devices (current generation smart watches).

References

Abandah G. and Khedher M. (2009). Analysis of handwritten Arabic letters using selected feature extraction techniques," Int'l J. of Computer Processing of Languages (IJCPOL). Vol. 22. No. 1. 49-73.

Abras, C., Maloney-Krichmar, D., and Preece, J. (2004). User-Centered Design, In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications.

Alotaiby, F., Alkharashi, I. A. and Foda, S. G. (2009). Processing Large Arabic Text Corpora: Preliminary Analysis and Results. Paper presented to the Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt.

Arab Standardization and Metrology Organization (ASMO). (1985). Standard 449: 7-Bit Arabic Code. Arab League, Cairo, Egypt.

Arab Standardization and Metrology Organization (ASMO). (1987). Standard 663: Arabic Keyboard Layout. Arab League, Cairo, Egypt.

Arabic reference at Ethnologue 17th ed. (2013). http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers#cite_note-12.

Bailey, R. W. (1996). Human performance engineering: Designing high quality, professional user interfaces for computer products, applications, and systems (3rd ed.). Upper Saddle River, NJ: Prentice Hall. 287.

Ballard, A., Shafer, P. and Sherwood T. (2006). Customization of tab-order functionality in internet applications, US Patent, no. 7111243.

Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D. and Furnas, G. (1994). Pad++: a zoomable graphical sketchpad for exploring alternate interface physics, Proc. ACM UIST 1994.

Bi, X., Li, Y., and Zhai, S. (2013) FFitts law: Modeling finger touch with Fitts' law. Proc. CHI 2013, ACM.

Bi, X., Smith, B. A. and Zhai, S. (2010). Quasi-qwerty softkeyboard optimization. In Proc. CHI 2010, ACM Press. 283-286.

Bishop, B. (1998). A history of Arabic Language. http://linguistics.byu.edu/classes/ling450ch/reports/arabic.html.

Black, A., (2008). User-centered Design, Design Council, URL: http://www.desigcouncil.org.uk.

Brown, C. M. (1988). Human-computer interface design guidelines. Norwood, NJ: Ablex Publishing.

Buxton, W.A.S. and Baecker, R.M. (1987). Human-computer interaction: a multidisciplinary approach, Morgan Kaufmann.

Card, S. K., Moran, T. P. and Newell, A. (1983): The Psychology of Human–Computer Interaction. Erlbaum, Hillsdale.

Cockburn, A. and Savage, J. (2003). Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods, In: Proceedings of the British Conference on Human Computer Interaction. HCI 2003. 87–102.

Computer Industry Almanac (2012). PCs In-Use Reached over 1.6B in 2011. URL: http://www.c-i-a.com/pr02012012.htm (February 1, 2012).

David, P. A. (1985). Clio and the Economics of QWERTY. American Economic Review 75(2): 332–337.

Dayao, A.B., Rafieyan, K., Lee, P.J., Graveman, R.F., Yacobi, Y., and Addeo, E.J. (1990). SuperForms: A Security-Enhanced Smart Electronic Form Management System, Proc. IEEE Global Telecommunications. Vol. 2.

Dix, A. J., Finlay, J., Abowd, G.D., and Beale, R., (2003). Human-Computer Interaction. Prentice-Hall, 3rd edition.

Dunlop, M.D. and Crossan, A. (2000). Predictive Text Entry Methods for Mobile Phones. Personal Technologies. Vol. 41. 134-143.

Dunlop, M. D. and Levine, J. (2012). Multidimensional Pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. CHI 2012.

Ehret, C. (2004) Interview by Word History Connected (WHC). http://worldhistoryconnected.press.illinois.edu/2.1/ehret.html.

El Batran, K. and Dunlop, M. (2013) Pareto Optimized Arabic Mobile Keyboard Layout. In: CHI 2013 Workshop on Grand Challenges in Text Entry. CHI 2013, Paris, France.

El Batran, K. and Dunlop, M. (2014). Improved form navigation on mobile devices. 6th International Conference of Computer Science & Information Technology. CSIT 2014. IEEE, Amman, Jordan, 96-99.

El Batran, K. and Dunlop, M. (2014). Enhancing KLM (keystroke-level model) to fit touch screen mobile devices. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. MobileHCI 2014. ACM, New York, NY, USA, 283-286.

El-Wakil, M. and Shoukry, A. (1989). On-line recognition of handwritten isolated Arabic characters, Pattern Recognition. Vol. 22(2). 97–105.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology. Vol. 47. No. 6. 381-391.

Gould, S. J. (1992). The Panda's Thumb of Technology. In Bully for Brontosaurus, 59–75. London: Penguin Books.

Gray, W. D., John, B. E., and Atwood, M. E. (1992). The Precis of Project Ernestine or an Overview of a Validation of GOMS. In Proceedings of the SIGCHI conference on Human factors in computing systems. ACM. 307- 312.

Grudin, J. (2007). A moving target: The evolution of human–computer interaction. In Sears, A. and Jacko, J. A. (2007). Human-Computer Interaction Handbook (2nd Edition). CRC Press.

Haggerty, B. Tarasewich, P. (2005). A new stylus-based method for text entry on small devices. 11th International Conference on Human-Computer Interaction HCII.

Harviainen, T.. Semitic languages (and the Phoenician language). http://phoenicia.org/semlang.html.

Hoggan, E., Nacenta, M., Kristensson, P. O., Williamson, J., Oulasvirta, A., and Lehtiö, A. (2013). Multi-touch pinch gestures: performance and ergonomics. In Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces. 219- 222.

Holleis, P., Otto, F., Hussmann, H., and Schmidt, A. (2007). Keystroke-level model for advanced mobile phone interaction. Proceedings of the SIGCHI conference on Human factors in computing systems. CHI 2007. 1505- 1514.

Holt, M.. (1999). The Afro-Asiatic language family. http://linguistics.byu.edu/classes/ling450ch/reports/afro-asiatic2.html.

Hoos, H.H. and Stutzle, T. Stochastic (2005). Local Search: Foundations and Applications. Morgan Kaufmann.

How, Y. and Kan, M. Y. (2005). Optimizing Predictive Text Entry for Short Message Service on Mobile Phones. HCII 2005.

Hunter, M., Zhai, S., and Smith, B. A. (2000). Physics-based graphical keyboard design. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems. CHI 2000. 157-158.

Idlebi, N. and Mrayati, M. (1990). Design Arabic keyboard layout based on statistical properties of Arabic characters, *Computers and the Arabic Language*.

Igarashi, T. and Hinckley, K. (2000). Speed-dependent automatic zooming for browsing large documents", Proc. ACM UIST 2000.

Isokoski, P., & Raisamo, R. (2004). Quikwriting as a multi-device text entry method. Proceedings of the Third Nordic Conference on Human-Computer Interaction - NordiCHI 2004. 105-108.

ITU (International Telecommunications Union) (2012). Measuring the Information Society URL: http://www.itu.int/ITU-D/ict/publications/idi/material/2012/MIS2012-ExecSum-E.pdf (November 9, 2012).

Jacko, J. A. and Sears, A. (2003). Human-Computer Interaction Handbook. Mahwah: Lawrence Erlbaum & Associates.

John, B. E.. (1990). Extensions of GOMS Analyses to Expert Performance Requiring Perception of Dynamic Visual and Auditory Information. In Proc. CHI'90. ACM Press. 107-115.

John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. ACM Transactions on Computer-Human Interaction (TOCHI), 3(4), 320-351.

Kaptelinin V. (1995). A comparison of four navigation techniques in a 2D browsing task, In: Proceedings of the Conference on Human Factors in Computing Systems (CHI 1995). ACM Press, New York. 282–283.

Khorshid, E., Alfadli, A. and Majeed, M. (2010). 'A new optimal Arabic keyboard layout using genetic algorithm', Int. J. Design Engineering. Vol. 3. No. 1.

Kieras, D., Using the Keystroke-Level Model to Estimate Execution Times. The University of Michigan, Unpublished Report, Online Version as of August 2006, http://www.pitt.edu/~cmlewis/KSM.pdf. 1993

Kristensson, P. O. (2007). Discrete and Continuous Shape Writing for Text Entry and Control. Ph.D. Dissertation, Linköping University, Sweden.

Kristensson, P. O., and Zhai, S. (2004). SHARK2: A Large Vocabulary Shorthand Writing System for Pen-Based Computers. In Proceedings of the Seventeenth ACM Symposium on User Interface Software and Technology, 43–52.

Luo, L. and John, B. E. (2005). Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model. CHI 2005. ACM Press. 1605-1608.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human–computer interaction. Human–Computer Interaction. Vol. 7. 91–139.

MacKenzie, I. S. (2002). Mobile text entry using three keys. Proceedings of the Second Nordic Conference on Human–Computer Interaction. 27–34.

MacKenzie, I. S. and Soukoreff, R. W. (1995). Theoretical upper and lower bounds on typing speeds using a stylus and soft keyboard. Behaviour & Information Technology. Vol. 14. No. 6. 370-379.

MacKenzie, I. S., and Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. *CHI 2003*.

MacKenzie, I. S., Zhang, S. X., and Soukoreff, R. W. (1999). Text entry using soft keyboards. Behaviour & Information Technology. Vol. 18. No. 4. 235-244.

MacKenzie, I.S. and Zhang, S. X. (2001). An empirical investigation of the novice experience with soft keyboards. Behaviour & Information Technology. Vol. 20. 411-418.

Ministry of communication and Information Technology (MCIT). (2013) Egypt ICT Sector Indicators in brief August 2013".

Mori, R., Matsunobe, T., and Yamaoka, T. (2003). A Task Operation Prediction Time Computation Based on GOMS-KLM Improved for the Cellular Phone and the Verification of that Validity, In Proc. ADC'03.

Myers, B. (1998). A brief history of human–computer interaction technology. Interactions 5(2):44–54, ACM Press.

Myung, R. (2004). Keystroke-Level Analysis of Korean Text Entry Methods on Mobile Phones, International Journal of Human-Computer Studies. Vol. 60. No. 5-6. 545-563.

Nationalencyklopedin. (2010). Världens 100 största språk 2010. The World's 100 Largest Languages in 2010.

Norman, D. A. (1988). The Psychology of Everyday Things. Basic Books, New York.

Oney, S., Harrison, C., Ogan, A., Wiese, J. (2013). ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In: Proc. 31st Annual SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, ACM, New York. 2799–2802.

Oulasvirta, A., Reichel A., Li W., Zhang Y., Bachynskyi M., Vertanen K., and Kristensson P. O. (2013). Improving two-thumb text entry on touchscreen devices. ACM CHI'13.

Pavlovych, A. and Stuerzlinger, W. (2004). Model for Non-expert Text Entry Speed on 12-button Phone Keypads. In Proc. CHI'04. ACM Press. 351-358.

Perlin, K. and Fox, D. (1993). Pad: an alternative approach to the computer interface, Proc. ACM SIGGRAPH 1993.

Pook, S., Lecolinet, E., Vaysseix, G. and Barillot, E. (2000). Context and interaction in zoomable user interfaces, In: Proceedings International Conference on Advanced Visual Interfaces (AVI '00). ACM Press, New York. 227–231.

Primary Human Language Families Map. (2013). http://en.wikipedia.org/wiki/File:Primary_Human_Language_Families_Map.png

Ruhlen, M. (1987). A Guide to the World's Languages.

Sears, A., Revis, D., Swatski, J., Crittenden, R., and Shneiderman, B. (1993). Investigating touchscreen typing: the effect of keyboard size on typing speed. Behavour & Information Technology. Vol. 12. No. 1. 17-22.

Sears, A. and Jacko, J. A. (2007). Human-Computer Interaction Handbook (2nd Edition). CRC Press.

Semitic languages. (2011). http:// http://en.wikipedia.org/wiki/File:Semitic_languages

Silfverberg, M., MacKenzie, I. S., & Korhonen, P. (2000). Predicting text entry speed on mobile phones. Proceedings of the CHI 2000 Conference on Human Factors in Com- puting Systems. New York: ACM.

Teo, L. and John, B. E. (2006). Comparisons of Keystroke-Level Model Predictions to Observed Data. In Extended Abstracts CHI'06. ACM Press. 1421-1426.

Toyama, K. (2010). Human–Computer Interaction and global development. Foundations and Trends in Human–Computer Interaction, 4(1), 1–79.

Tran, J. J., Trewin, S., Swart, C., John, B. E. and Thomas, J. C. (2013). Exploring pinch and spread gestures on mobile devices. MobileHCI 2013. 151-16.

Tudor, L. G., Muller, M. J., Dayton, T., and Root, W. (1993). A participatory design technique for high-level task analysis, critique, and redesign: The CARD method. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting, Santa Clara, CA: HFES. 295-299.

Yannou, B. and Hossenlopp, P. (2000). Clavier alphanumérique ergonomique. Brevet d'invention. no 98-04588, numéro de publication 2 777 222, Ecole Centrale Paris, Chatenay – Malabry, France.

Zhai, S., Hunter, M., and Smith, B. A. (2000). The Metropolis keyboard: An exploration of quantitative techniques for virtual keyboard design. Proceedings of the ACM Conference on User Interface Software and Technology - UIST 2000, 119-128.

Zhai, S., Hunter, M., and Smith, B. A. (2002). Performance Optimization of Virtual Keyboards. Journal of Human Computer Interaction. Vol. 17. No. 2-3. 229-270.

Zhai, S., and Kristensson, P. O. (2003). Shorthand Writing on Stylus Keyboard. In Proceedings of the ACM Conference on Human Factors in Computing Systems, 97–104.

Appendix A: Fitts' Law

Fitts's law is a model of human movement primarily used in human–computer interaction and ergonomics that predicts the time required to rapidly move to a target area, it is a function of the distance to the target and the size of the target. In other words, formal relationship that models speed/accuracy tradeoffs in aimed movements (Accot and Zhai, 1997). It was proposed in 1954 by Paul Fitts' to model the act of pointing, either physically, touching an object with a hand or finger, or virtually, by pointing to an object on a computer, mobile or tablet screen using a pointing device or finger.

Fitts' law proved its superiority although Fitts only published two articles on his law (Fitts 1954, Fitts and Peterson 1964). Yet there are hundreds of subsequent studies related to it in the human–computer interaction (HCI) literature. However, the first HCI application of Fitts's law was by Card, English, and Burr (1978), who used the index of performance (IP), defined as 1⁄b, to compare performance of different input devices, with the mouse coming out on top. This early work, according to Stuart Card's biography, "was a major factor leading to the mouse's commercial introduction by Xerox".

Moreover, Fitts's law has been applied to tasks where the user must position a mouse cursor over an on-screen target, such as a button or other widget. Fitts's law models both point-and-click and drag-and-drop actions. Dragging has a lower IP associated with it, because the increased muscle tension makes pointing more difficult. Fitts's law has also been shown to model target-directed hand and head motions in a virtual environment (So and Cheung, 2002) (So et al., 1999), head-tracking performance (So and Griffin, 2000), and hand and foot movement times (Errol and Hoffmann, 1991).

In this appendix, an introduction on Fitts' Law will be illustrated. The first section will cover the model and its parameters. The second section will focus on the mathematical details of the model. The third section will give a brief derivation of the model. The fifth section will show the model's success and implications. The final section will provide a brief summery.

A.1 Model

The Shannon formulation is one of the various mathematical formulas of Fitts's law, which was proposed as a common form by Scott MacKenzie. It was named for its resemblance to the Shannon–Hartley theorem for movement along a single dimension:

$$T = a + b\ \log_2\left(1 + \frac{2D}{W}\right)$$

where:

- T is the average time taken to complete the movement. It may also be referred also as MT (mean movement time).
- a represents the incorporating reaction time and/or the time required to execute the operator.
- b stands for the inherent speed of the device.
- D is the distance from the starting point to the center of the target. It may also be referred also as A (amplitude of the movement).
- W is the width of the target measured along the axis of motion. It can allow error tolerance in the final position, providing the final point falls within ±W⁄2 of the target's center.

From the equation, the constants a and b (intercept coefficients and slope respectively) in the prediction equation can be determined experimentally by fitting a straight line to measured data, typically using linear regression. As seen, speed is inversely proportional to accuracy, where more time is required to acquire targets that are smaller and/or further away.

Fitts' law has certain parameters that needed to be taken in consideration regarding the formation of time prediction equation:

- It applies only to movement in a single dimension and not to movement in two dimensions.

- It describes simple motor response of the human hand, usually implemented for a mouse cursor.
- It describes untrained movements, not movements that are executed after months or years of practice.
- Buttons and other GUI controls should be a reasonable size; it is relatively difficult to click on small ones.

A.2 Mathematical details

Fitts' law is often used to build a prediction model with the movement time (MT) to complete point-select tasks as the dependent variable:

$$T = a + b\, ID,$$

where

$$ID = \log_2\left(\frac{D}{W} + 1\right).$$

As mentioned previously, the constant b is the inherent speed of the device measured in time/bit, while the constant a can be thought of as incorporating reaction time and/or the time required to and/or the time required to execute the operator. The values of a and b changes accordingly to the conditions under which pointing is done, where constants a and b associated for using a stylus are different than the constants a and b associated for using a mouse or finger for pointing.

The logarithm in Fitts's law is called the ID (Index of Difficulty). It is the measurement of the movement task's difficulty using information theory by the metric "bits" (MacKenzie, 1995). Slightly different from the Shannon formulation is the original formulation by Fitts:

$$ID = \log_2\left(\frac{2D}{W}\right).$$

The factor "2" in the logarithm was added by Fitts as an arbitrary adjustment to ensure that ID was greater than zero for the range of experimental conditions employed in his experiments (Fitts, 1954). The "2" increases the index of difficulty by 1 bit for each task but has no effect on the MT-ID correlation or on the slope of the regression line equation (MacKenzie, 1992). The constant "1" in Shannon's original equation made a different from Fitts's original form, especially for low values of the ratio D/W. The Shannon form has the advantage that the ID is always non-negative, and has been shown to better-fit measured data.

An index of performance IP can be defined to characterize how quickly pointing can be done, independent of the particular targets involved Zhai (2002). It measures the pointing capability of an input device by the metric "bits/time". IP eases the comparison of different input devices with respect to their pointing capability. The index of performance (IP) can be calculated by dividing a task's index of difficulty by the observed movement time (MacKenzie, 1992).

A.3 Derivation

There is an easy way to derive Fitts' law from a simple model. The below derivation is proposed by Drewes in 2010. The model, often called discrete-step-model, is standard knowledge in Fitts' law research, but is reinvented here, as it was not possible to find the first author who presented it. In this model the pointer approaches the target in steps. In each step the pointer aims to the target center and reaches a position within an error circle from where the next step starts. Each step consumes the same amount of time and brings the pointer gradually closer to the target. The process ends when the pointer is inside the target. See figure A.1 for an illustration.

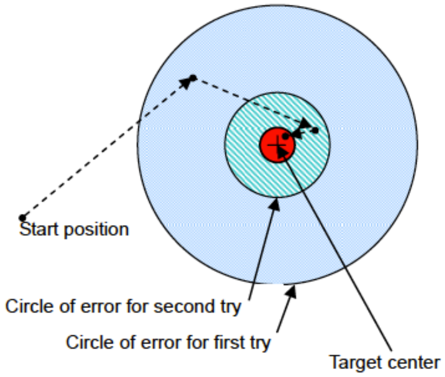


Fig A.1 Step-wise movement towards target (Drewes, 2010)

Let the distance to the target at each stage be A_i with the initial distance $A_0 = A$. After each step, the average distance to the center of the target A_{i+1} is a constant fraction λ of the distance Ai at the beginning of the step.

$$A_{i+1} = \lambda \cdot A_i$$

And consequently:

$$A_i = \lambda^i \cdot A$$

The process stops after n steps when the distance to the target center is less than the radius R of the target:

$$A_n = \lambda^n \cdot A < R$$

From this derives:

$$n = \log (R / A) / \log (\lambda)$$

Each step takes a fixed time τ and there will be some initial time a for the brain to get started. The total time t to reach the target is:

$$t = a + \tau \cdot n = a + \tau \cdot \log (R / A) / \log (\lambda)$$

$$t = a + b \cdot \log (A / R)$$

Where $b = - \tau / \log (\lambda)$. As the pointer gets closer to the target with each step, λ is smaller than 1 and log (λ) is negative so b is positive.

A.4 Summary

The main focus behind this appendix was introducing Fitts' law in general. It was concluded Fitts's law remains one of the few hard, reliable human–computer interaction predictive models that predict the time required to rapidly move and/or point to a target area. In next appendix publication done throughout the research will be introduced.

A.5 References

Card, S. K., English, W. K., and Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. Ergonomics. Vol. 21. No. 8. 601–613.

Card, S. K., Moran, T. P., and Newell, A. (1983): The Psychology of Human–Computer Interaction. Erlbaum, Hillsdale.

Drewes, H. (2010). Only one Fitts' law formula please!, CHI '10 Extended Abstracts on Human Factors in Computing Systems, April 10-15, Atlanta, Georgia, USA.

Errol, R., Hoffmann (1991) "A comparison of hand and foot movement times". Ergonomics. Vol. 34. No. 4. 397-406.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology. Vol. 47. No. 6. 381–391. (Reprinted in Journal of Experimental Psychology: General. Vol. 121. No. 3. 262–269.

Fitts, P. M. and Peterson, J. R. (1964). Information capacity of discrete motor responses. Journal of Experimental Psychology. Vol. 67. No. 2.103–112.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human–computer interaction. Human–Computer Interaction. Vol. 7. 91–139.

MacKenzie, I. S. (1995). Movement time prediction in human-computer interfaces. In R. M. Baecker, W. A. S. Buxton, J. Grudin, & S. Greenberg (Eds.), Readings in human-computer interaction (2nd ed.), 483-493. San Francisco: Kaufmann.

So, R. H. Y., Cheung, K. M. and Goonetilleke, R. S. (1999) "Target-directed head movements in a head-coupled virtual environment: predicting the effects of lags using Fitts' law". Human Factors. Vol. 41. No. 3. 474–486.

So, R. H. Y. and Griffin, M. J. (2000) "Effects of target movement direction cue on head-tracking performance". Ergonomics. Vol. 43. No. 3. 360–376.

So, R. H. Y. and Cheung, K. M., (2002) "Combined and interacting effects of hand and head movement lags on discrete manual performance in a virtual environment." Ergonomics. Vol. 45. 105–123.

Zhai, S. (2002). On the Validity of Throughput as a Characteristic of Computer Input, IBM Research Report RJ 10253, Almaden Research Center, San Jose, California.

Appendix B: Publications

B.1 Extended abstract for CHI workshop 2013

An extended abstract submitted to CHI workshop 2013 - Grand challenges of text entry.

- Pareto Optimized Arabic Mobile Keyboard Layout.

B.2 Short paper for CSIT 2014

A Short paper submitted to CSIT 2014 - The 6th International Conference on Computer Science and Information Technology.

- Improved form navigation on mobile devices: A case study with the Egyptian Agricultural Survey.

B.3 Short paper for MobileHCI 2014

- Enhancing KLM (Keystroke-Level Model) to Fit Touch Screen Mobile Devices

Pareto Optimized Arabic Mobile Keyboard Layout

Karim El Batran and Mark Dunlop
Computer and Information Sciences, University of Strathclyde.
Richmond St, Glasgow, G1 1XH, Scotland, UK.
karim.el-batran@strath.ac.uk
Mark.Dunlop@cis.strath.ac.uk

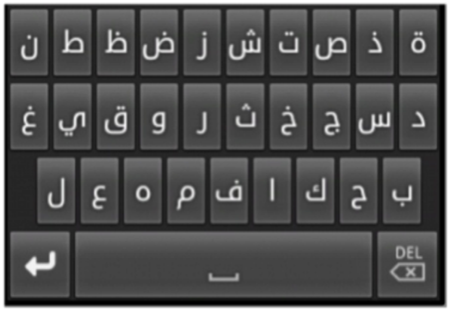


Figure 1. Pareto optimized Arabic keyboard layout

CHI'13 Text Entry Workshop, April 28, 2013, Paris, France.

Abstract

This paper presents a new design of an Arabic keyboard layout for effective text entry on touch screen mobile phones. Our approach is based on Pareto front optimization using three metrics: minimizing finger travel distance in order to maximize speed, minimizing neighboring key error ambiguities in order to maximize the quality of spell correction, and maximizing familiarity for less technologically literature users through approximate alphabetic sorting. In our short user studies, the new layout showed an observed improvement in typing speed in comparison to a common Arabic layout. We believe the opportunity is now ripe to research new optimized keyboard designs where there is less usage experience than QWERTY has in mainstream Western European languages. Pareto optimization can produce high quality keyboards for alphabet based languages that could give real benefits where there is less reluctance to change from Qwerty.

Author Keywords

Touch-screen keyboard design optimization.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies.

General Terms

Performance, Design, Human Factors.

Introduction

There has been a long history of work looking at speed and accuracy in text entry for English [e.g. 9, 12, 16 and 17]. For example, it has long been known that smaller keys can impact on accuracy [14], speed or both [15]. The lack of physical touchable targets in touch-screen smartphones accentuates the “fat finger” problem as users fingers have low accuracy on these screens [7]. Despite Arabic being the seventh most written language (4.23% world population [13]) there has been very little work on Arabic text entry. There is also less keyboard standardization than for English and, technological backgrounds and are less experienced using a particular keyboard layout than is typical for English. We believe this gives a perfect opportunity to design a keyboard layout that would not suffer the uptake problems of optimized English keyboards.

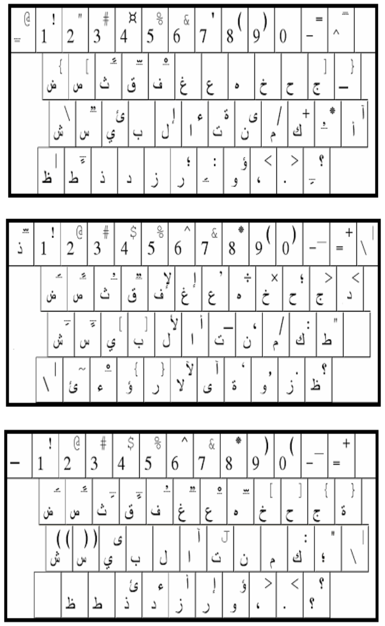


Figure 2. ASMO 663, PC and Apple Arabic layouts

Optimizing keyboard layouts

For English and many languages the Qwerty layout (and variants such as AZERTY) was adopted on touch screen phones through familiarity. However, touch keyboards have very small keys and no tactile feedback leading to increased error rates on touch screen phones than on physical keyboards [1]. Alternatives keyboard layouts, including Dvorak [5], have not been successful for many reasons, but largely because of the high initial learning curve. Bi, Smith and Zhai introduced an alternative layout to improve Qwerty while maintaining familiarity [4]. They rearranged the Qwerty layout by shuffling keys at most one position from their original location to achieve a quasi-optimized Qwerty variant in the hope of improving acceptance and take-up by users.

Optimization has also been extended to also attempt to reduce the likelihood that hitting a neighboring key will result in a valid dictionary word, and thus improve error correction and accuracy. Dunlop and Levine introduced the concept of badgrams – pairs of letters that when one is substituted for another often result in valid words (e.g. *I* and *O* on a QWERTY keyboard leads to words such as *in/on*, *sin/son*, and *fir/for* being differentiated by very small keyboard distances) [6].

Arabic text entry

The common Arabic desktop keyboard layout is derived from the 1970s Arabic typewriter. However, there is less standardization than for English QWERTY with different keyboard variations in use on different devices in the same country. The Arab Standardization and Metrology Organization (ASMO) developed the standard Arabic desktop keyboard layout supporting the 7-bit Arabic character code [2, 3]. The market however adopted the now widely used Microsoft PC layout, while Apple use an alternative for iOS and OS X. Figure 2 shows these three main variants of the Arabic layout. While not drastically different this will have an impact on users being more open to variations.

There has been no research on optimizing Arabic text entry on mobiles. However, on desktops, Idlebi and Mrayati designed an efficient Arabic keyboard based on statistical approach [10]. Khorshid used a genetic algorithm to optimize the keyboard layout for typing speed. In the genetic algorithm, a fitness function was used taking into consideration key distance, finger used, and hand alternation targeting ten-finger typing. Their studies showed a 36.3% speed improvement over the present PC Arabic keyboard [11].

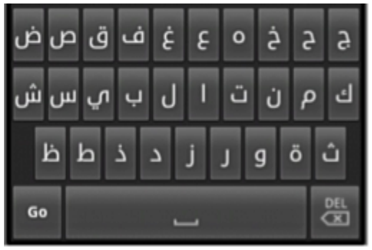


Figure 3. Apple Arabic keyboard layout

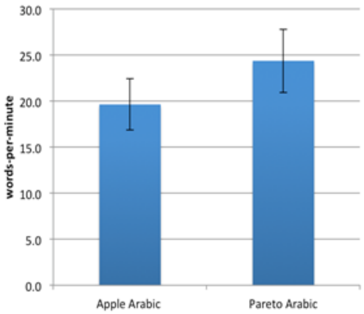


Figure 4. Average estimated words-per-minute

PARETO OPTIMIZATION OF ARABIC

Pareto optimization is a variant of local neighborhood search adapted for multiple criteria searching [8]. We used a three-dimensional Pareto front optimization [6] that attempts to optimize for (a) speed of text entry, (b) error correction and (c) familiarity.

The development of a keyboard requires analysis of a large representative text corpus – no such common corpus exists for Arabic. We analyzed 2004 issues of the Arabic newsletter Watan containing articles from fields such as culture, economy, news, religion, and sports. The corpus contains 126,913 unique words (34,009,607 occurrences). We calculated bigram and neighbor ambiguity (badgrams) information. The bigram analysis resulted in a weight for each two-letter Arabic bigram – we recorded occurrence information between the 36 characters and between each character and space to give 37*37 pairs. The top four key pair probabilities were = ن_ ,0.071 = ي_ 0.060 = _ا ,0.068, and 0.054 = لا (where _ represents space). Compared to English there were very few badgrams for Arabic. We believe that this is a result of the reduced role of vowels in written Arabic. The most common badgrams were "ام" and "يم", however we only saw 32 badgrams in total (compared to 325 for English), where "ام" represents only 0.003% of the bigram occurrences compared to 0.017% for "AE" in English. A function that measured similarity between novel layouts and Qwerty was included by Dunlop and Levine as a third metric metric to attempt to minimize deviation from Qwerty. Given the lower levels of computer and keyboard literacy in Country together with the variation in desktop layouts, we felt that optimizing to a standard keyboard was less suitable. As such, our third metric was familiarity with a column-major alphabetic keyboard, with the alphabet running from right to left.

An initial set of randomly generated keyboards formed an initial set of potential solutions. This set then went through 2000 iterations in which each keyboard had a small number of keys swapped. If the new keyboard was better on ANY metric than an existing solution, it was added to the set; if it was at least as good on ALL metrics then it dominates the existing one, which is discarded. This leads to a Pareto front – a set of dominant solutions on a 3D surface. At the end of the process we normalized the space so that each dimension ranged from 0 to 1 and took a good compromise design from the center of the Pareto front.

INITIAL USER EVALUATION

We evaluated our optimized layout against the Apple standard with HTC Desire smartphones (Fig 1,3). We recruited 20 participants aged 20-40 of whom 60% were familiar with touch screens and 85% were PC users. In a balanced study, users were asked to enter 3 phrases as practice then 20 test phrases on each keyboard. Phrases were translations from the English standard MacKenzie and Soukoreff set [18].

RESULTS

The results in figure 4 show that users typed at an estimated 24.4 words-per-minute on average using the Pareto optimized Arabic keyboard layout, but 19.6 wpm using the Apple layout[1]. Despite the relatively short study this shows an indicative speed advantage for our keyboard (2- tailed paired t-test, n=20, p<0.1) over

[1] Our system failed to record wpm data from some phones in our study. Times for these were estimated based on server logs and estimates crosschecked where full data was available.

the standard keyboard for entry by educated users in Country. Furthermore, the majority of users preferred our layout due primarily to the near alphabetic key distribution. However, some users noted that more time is needed to practice the optimized keyboard layout. We now plan to extend the study to groups with different IT usage experience and education levels.

DISCUSSION

This paper has introduced a new three-dimensionally optimized Arabic mobile keyboard layout. We observed an indicative speed improvement over a standard layout in a short study. Users appreciated the near-alphabetic layout. This is in strong contrast to study in English where considerable practice is needed to overcome the heritage of Qwerty.

Based on this work we believe there is a strong opportunity to introduce new optimized keyboards for alphabetical laguages in less technically developed nations (e.g. Arabic and Swahili or even English, French, and Portuguese in Africa and Middle East).

Acknowledgements

To be added on non-anonymous version

References

[1] Allen, J. M., McFarlin, L. A. and Green, T. An In-Depth Look into the Text Entry User Experience on the iPhone. In Proc. 52nd HFES (2008), 508-512.

[2] Arab Standardization and Metrology Organization 1985. Standard 449: 7-Bit Arabic Code.

[3] Arab Standardization and Metrology Organization 1987. Standard 663: Arabic Keyboard Layout.

[4] Bi, X., Smith, B. A. and Zhai, S. Quasi-qwerty soft keyboard optimization. CHI 2010.

[5] David, P.A. Clio and the Economics of QWERTY. American Economic Review 75 (1985).

[6] Dunlop, M. D. and Levine, J. Multidimensional Pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking, CHI 2012.

[7] Holz, C. and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. CHI 2010.

[8] Hoos, H.H. and Stutzle, T. Stochastic Local Search: Foundations and Applications. Morgan Kaufmann (2005).

[9] Hunter, M., Zhai, S., and Smith, B. A. Physics-based graphical keyboard design. CHI 2000.

[10] Idlebi, N. and Mrayati, M. Design Arabic keyboard layout based on statistical properties of Arabic characters, Computers and the Arabic Language, 1990.

[11] Khorshid, E., Alfadli, A. and Majeed, M. 'A new optimal Arabic keyboard layout using genetic algorithm', Int. J. Design Engineering, 3(1) (2010).

[12] Lewis, J. R., Kennedy, P. J. and LaLomia, M. J. Development of a Digram-Based Typing Key Layout for Single-Finger/Stylus Input. HFES 1999.

[13]http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[14] MacKenzie, I. S. and Soukoreff, R. W. Theoretical upper and lower bounds on typing speeds using a stylus and soft keyboard. Behaviour & Info Tech, 14(6) 1995.

[15] Sears, A., Revis, D., etl al. Investigating touchscreen typing: the effect of keyboard size on typing speed. Behavour & Info Tech 12(1) 1993.

[16] Zhai, S., Hunter, M., & Smith, B. A. The Metropolis keyboard: An exploration of quantitative techniques for virtual keyboard design. UIST 2000.

[17] Zhai, S., Hunter, M., and Smith, B. A. Performance Optimization of Virtual Keyboards. Journal of Human Computer Interaction, 17(2 & 3) 2002.

[18] MacKenzie, I. S., & Soukoreff, R. W. Phrase sets for evaluating text entry techniques. CHI 2003.

Improved Form Navigation on Mobile Devices

A case study with the Egyptian Agricultural Survey

Karim El Batran
Computer and Information Sciences Dept.
University of Strathclyde
Glasgow, Scotland UK
karim.el-batran@strath.ac.uk

Mark Dunlop
Computer and Information Sciences Dept.
University of Strathclyde
Glasgow, Scotland UK
Mark.Dunlop@cis.strath.ac.uk

***Abstract*— in this paper a study was done regarding improving form control and navigation on mobile devices. An observational study of census counters for the Egyptian Agricultural Survey was conducted. This country-wide survey is currently conducted by professional counters using large (100x35 cm double sided) complex paper forms that require manual transcription. Computerization would be beneficial in terms of accuracy and duplication of effort. However there are considerable challenges in reducing the size of the forms to fit mobile devices. Furthermore counters typically have low technological experience. Based on our observations we developed two prototypes: one using traditional form tabs, the other pan-and-zooming. Results from initial user tests showed the pan-and-zoom interface was both faster and had a lower perceived effort.**

Keywords*— Mobile Devices; HCI (Human Computer Interaction); Usability; LTBU (Low Technological Background User); small-screen devices navigation and Form display.

I. Introduction

A form is a document (printed or electronic) with spaces in which to write or enter data. Forms are used to increase the uniformity and efficiency of handling routine tasks, to ensure that tasks are completed and that information can be found when needed [4]. The quality of digital form interfaces depends primarily upon three factors. One of these factors is the clarity of the design and visual representation of the screens [3]. In the bulk of current form design practice the element, layout, and functionality of user interfaces is intended for casual users who access the forms on an irregular basis and prefer simple interfaces rather than more powerful alternatives. This has led to the adoption of basic point and click interaction model in which navigation and control was accomplished by selecting user interface elements with a pointing device such as a mouse [1]. Today, new techniques are widely used to interact with touchscreen interfaces, particularly on mobile devices, yet the interaction style for forms has not been re-evaluated.

Completing complex forms is both time consuming and error prone with users making incorrect selections and spending considerable time navigating the form. Traditional scrollbars were used in navigating through large forms so that the large information space could fit on screen. Igarashi and Hinckley [5] have identified that one of the major two limitations with using traditional interaction is that users have to shift their focus between the document and the scrollbar. They suggest that this may increase the operational time and may cause a significant attentional overhead. Furthermore, in large documents small scrollbar movements can cause a large movement of the document that disorients users.

To counter this visual blurring and to reduce physical navigational workload, there were two techniques. First, the tabbing technique, which converts the form into several separate parts. Second, pan and zoom interfaces, typical of image viewing applications, view the whole document as one navigable screen [2, 6]. Successive zooming in and out by the user gradually expands and contracts the form contents while panning changes the section of the form to be displayed. Zooming and panning technique used for document and map navigation tasks have proved to achieve user's acceptance and improvement in the overall performance and are now widespread on tablets and phones. However, their use in complex form filling has yet to be fully explored.

Here we initially introduce a case study of the Egyptian Agricultural Census. We then describe an observational study conducted with census counters and two versions of a tablet based solution, one using traditional tabbing and one pan-and-zoom. Finally, we discuss user studies conducted to assess two iPad interfaces for navigating and completing the complex forms. Finally, we conclude and discuss future work.

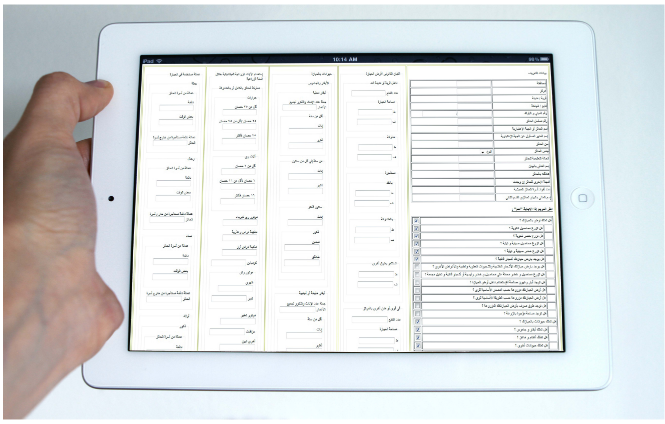


Fig. 1. View of the entire form using zoom technique.

II. Case Study

The Egyptian Agricultural Census is a comprehensive inventory of the national economic structure of agriculture in a specific time period (one agricultural year). It is done under the supervision of the state by collecting accurate information about agricultural production units (agricultural holdings such as number of agricultural holdings, areas, uses, geographically characterization and categorization, status of irrigation and drainage, number of livestock and poultry, agricultural machinery and farmers), the features and characteristics of such possessions. It is also used to measure the direction and rates of change compared to the results of previous censuses and current statistics, in order to discern characteristics and development of the economic structure in the agricultural sector. The most important purpose of implementing agricultural census is to rely on its results as a basis for planning by economists, social researchers and also draw a final picture that describes the relationship between different economic sectors. In next section, we introduce an observational study conducted on the Egyptian Agricultural Census.

III. The Observational Study

The EAC (Egyptian Agricultural Census) deals with an inventory of all possessions located in villages and cities of the Nile Valley, as well as possessions located in desert governorates and land belonging to the reconstruction organization.

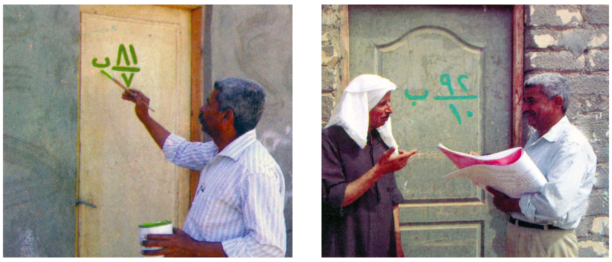


Fig. 2. Numbering buildings and census interviews while completing paper forms.

As shown in figure 2, the census is conducted through personal interviews done by counters with the holders in their place of residence after numbering the buildings to identify them easily in villages and small towns. However, because in big cities numbering is difficult, the counters follow the method of inventory by walking through agricultural basins and taking landholders addresses in order to reach them.

The EAC is conducted through two phases. First phase, a sketch is set at the beginning for all villages and towns nationwide, the buildings are then numbered and a visit is made to all the holders to collect initial data. Second phase, a visit is made to all the families to collect more precise data including personal information and other detailed data regarding their possession. All gathered data are written in paper forms. The preliminary stage of the EAC contains four steps; issuing executive decisions to conduct the EAC, preparing the census budget and adopting it, choosing the field supervisors, engineers, agents and participants in the governorates and training workers on the agricultural census. A trial census is done in order to measure the efficiency of the forms used, revise the method of data collection, identify problems and obstacles and observe the response of holders to questions on the census. This trial census was held in three selected governorates representing the regions of Upper and Lower Egypt.

There are certain obligatory rules that must be adhered to by counters and the head-of-counters. First, the allocation of one head-of-counters to each five counters. Second, taking into account the equitable distribution among counters and heads-of-counters in areas of operation. Third, the revision of the numbering that took place in the first phase, buildings not mentioned should be documented by writing a notice. Forth, buildings must be visited in the same order of the records and all the holders contained in the records must be visited. Finally, the heads-of-counters accept records after being reviewed precisely.

The following flow chart illustrates how the process of the census and data collection is done.

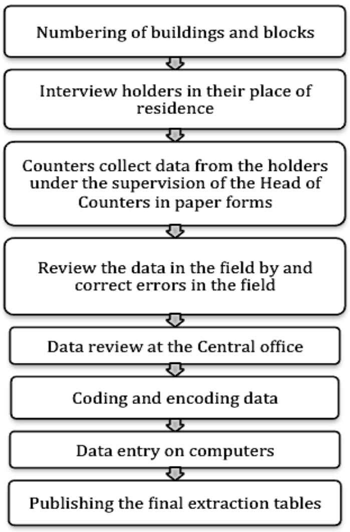

Numbering of buildings and blocks
Interview holders in their place of residence
Counters collect data from the holders under the supervision of the Head of Counters in paper forms
Review the data in the field by and correct errors in the field
Data review at the Central office
Coding and encoding data
Data entry on computers
Publishing the final extraction tables

Fig. 3. Egyptian Agricultural Census Process Flow.

In our observations and discussions with senior counters, we found most counters could be categorized as Low Technological Background users (LTB). LTB users are categorized as having low to zero levels of experience dealing with computerized devices such as PCs, laptops, tablets, smart-phones and/or cell phones. For reference, the percentage of computer users was approximately 23% in Egypt in July 2010. This percentage increased to 42.78% in July 2013 [7].

IV. Computerization Challenge

Although, there is technological progress in information-gathering systems and data collection, the Egyptian Ministry of Agriculture and Land Reclamation still collects the data manually on large (double sided 100x35 cm) paper forms contained in books of 40 forms, as in the right image in figure 1. The Ministry then transfers the books physically to the headquarters, and then enters the data to a database using “Data Entry” employees separate from the counters – primarily because of the low technical background typical of the counters. This double entry has been identified as affecting the accuracy of data and increasing the cost of the census. However, introducing more computerized data collection using electronic forms leads to interesting questions such as: What is the attitude of the LTB users towards computerizing the forms? How will the change of size from large forms to tablets affect interaction? In particular will the need to navigate through the form affect external working memory of the counters? Do pan and zoom techniques better support their working memory than traditional tabbing interfaces? Will computerizing the process help in decreasing the overall time, increasing its accuracy and decreasing the cost?

V. Initial Comparative User Studies

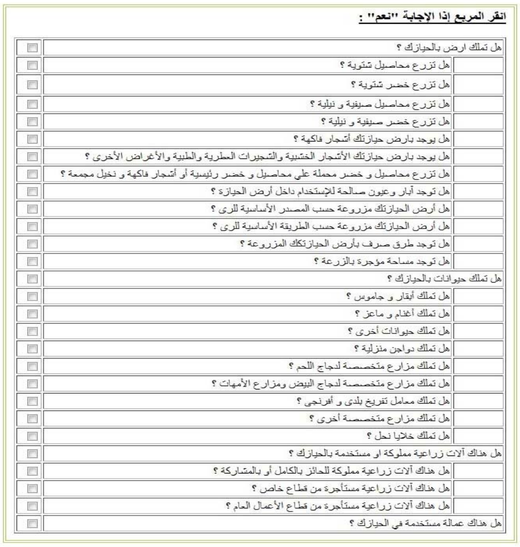
Our observational studies highlighted the complexity of forms completed, their physical size and the low technical background of census counters. Here, we compare using two interfaces, one based on tabbing and one on zooming, for completing forms on tablets to assess which interface style is easier to use for the counters given their experience.

A. System Description

The system is aimed to minimize the time, which the counter takes interviewing the holder. With our system, counters initially answer a number of yes/no questions to generate the form elements required. Afterwards, the counter begins filling and submitting the form.

B. System Design

The form is designed using HTML5 and JavaScript to be displayed on an Apple iPad 2 using the standard Safari browser (9.7” screen with resolution of 1024x768). It is based on two techniques: selecting and typing. The form is composed of a combination of checkboxes and textboxes and was designed to closely resemble the original paper version in structure and individual components



انقر المربع إذا الإجابة "نعم" :

هل تملك ارض بالحيازك ؟ □
هل تزرع محاصيل شتوية ؟ □
هل تزرع خضر شتوية ؟ □
هل تزرع محاصيل صيفية و نيلية ؟ □
هل تزرع خضر صيفية و نيلية ؟ □
هل يوجد بارض حيازتك أشجار فاكهة ؟ □
هل يوجد بارض حيازتك الأشجار الخشبية والشجيرات العطرية والطبية والأغراض الأخرى ؟ □
هل تزرع محاصيل و خضر محملة علي محاصيل و خضر رئيسية أو أشجار فاكهة و نخيل مجمعة ؟ □
هل توجد آبار وعيون صالحة للإستخدام داخل أرض الحيازة ؟ □
هل أرض الحيازتك مزروعة حسب المصدر الأساسية للرى ؟ □
هل أرض الحيازتك مزروعة حسب الطريقة الأساسية للرى ؟ □
هل توجد طرق صرف بأرض الحيازتكك المزروعة ؟ □
هل توجد مساحة مؤجرة بالزرعة ؟ □
هل تملك حيوانات بالحيازك ؟ □
هل تملك أبقار و جاموس ؟ □
هل تملك أغنام و ماعز ؟ □
هل تملك حيوانات أخرى ؟ □
هل تملك دواجن منزلية ؟ □
هل تملك مزارع متخصصة لدجاج اللحم ؟ □
هل تملك مزارع متخصصة لدجاج البيض ومزارع الأمهات ؟ □
هل تملك معامل تفريخ بلدى و أفرنجى ؟ □
هل تملك مزارع متخصصة أخرى ؟ □
هل تملك خلايا نحل ؟ □
هل هناك آلات زراعية مملوكة او مستخدمة بالحيازك ؟ □
هل هناك آلات زراعية مملوكة للحائز بالكامل أو بالمشاركة ؟ □
هل هناك آلات زراعية مستأجرة من قطاع خاص ؟ □
هل هناك آلات زراعية مستأجرة من قطاع الأعمال العام ؟ □
هل هناك عمالة مستخدمة في الحيازك ؟ □

Fig. 4. 28 (yes/no) questions.

The paper Agricultural Census form is divided into 5 main sections containing 666 fields to be filled in with data (not all fields are mandatory). Section 1 contains 16 mandatory questions regarding the holder’s information; Section 2, data regarding the possession of land (420 fields); Section 3 data about livestock and animals (100 fields); Section 4 concerns Agricultural machinery (63 fields); and the final section, information regarding labor used (57 fields). These five sections are reflected in the electronic form. The first section is visible to enter in the data. The other four are optional and initially hidden, set to visible according to the selection of their checkbox on the overview section.

First, the counter enters the essential data of the holder’s personal information. The counter asks the holder 28 yes/no questions as shown in figure 4; these questions generate the entire form. There are four main questions representing the remaining four main sections of the form. Each main question contains a number of sub questions, which represents a set of required information to be given by the holder and filled by the user (counter). This part represents the first section of the form. Subsequently, the counter begins filling in the form. Later, after editing, updating fields and making sure that all fields are completed and correct, the user submits the form.

C. User Evaluation

We conducted a usability study in which we compared the zoom and tab interfaces in terms of task-completion times and workload. We recruited 20 counters. Reflecting the typical counter our subjects were all male, aged 25-42 and bachelors graduates. Their fields of study varied greatly, but not computer science; none were familiar with tablets or extensive PC users, but all were familiar with the original paper version of the Agricultural Census.

As mentioned before, we are comparing two navigation techniques; the pan and zoom technique against the traditional tabbing technique. In the pan and zoom technique, the method of zooming in on the screen image includes sliding your thumb and index finger outward across the screen.

This method allows the user to zoom in on specific areas in the form for edit or update. Restoring the entire form requires from the user to slide his thumb and index finger inward across the screen in a pinching motion. This enables the user to view the whole form in one screen. In the traditional tabbing technique, the form is viewed and divided into multi-tabs view. Each section is displayed in a separate tab. This method allows the user to view each section separately.

Figures 1 and 5 show the two interface versions. Users were asked to complete two scenario forms on each device. The dependent variables were task-completion time (in seconds), interface preference, and subjective workload. The workload was measured with the NASA Task Load Index questionnaire (NASATLX). Tasks and order of presentation of interfaces were randomly balanced.

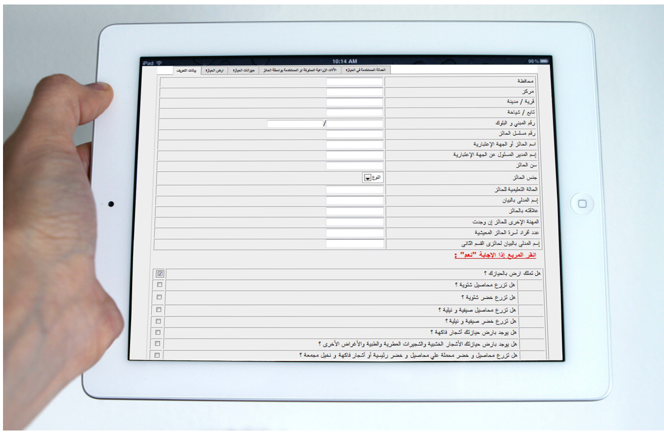


Fig. 5. View of the form using tabbing technique.

D. Results

Our results show that it took our participants about 31 minutes on average to complete the tasks using zooming technique, but 35 minutes on average using tabbing technique. This showed significantly lower task completion time ($p<0.05$, paired t-test, df = 19) as shown in figure 6. Moreover, NASA TLX forms were completed after each interface and showed indicatively lower workload for effort ($p<0.1$) using the zooming technique.

Finally, users were asked at the end of the study “Did you prefer the computerized version of the form, which navigation technique do you prefer and why”. All participants were satisfied using the electronic form. Fourteen of the twenty users favored the zooming technique. Their comments were clear regarding the ability of viewing the whole form rather than hopping through tabs. A couple of them suggested adding the ability of the user to determine the uncompleted sections or fields in the form.

While the results show that the zooming technique was significantly faster than tabbing and we attempted to have representative users, the results are based on a relatively small sample of participants doing artificial tasks in a laboratory setting. Longer field-based trials are needed to confirm the results and to assess fully whether a zooming based tablet application could improve on paper-based forms.

VI. Conclusion

We have presented a study of professional form use through an observational study of census counters. We then developed two tablet-based forms to replace paper forms. We introduced a new approach to improve form navigation for form filling on mobile devices using pan-and-zoom techniques.

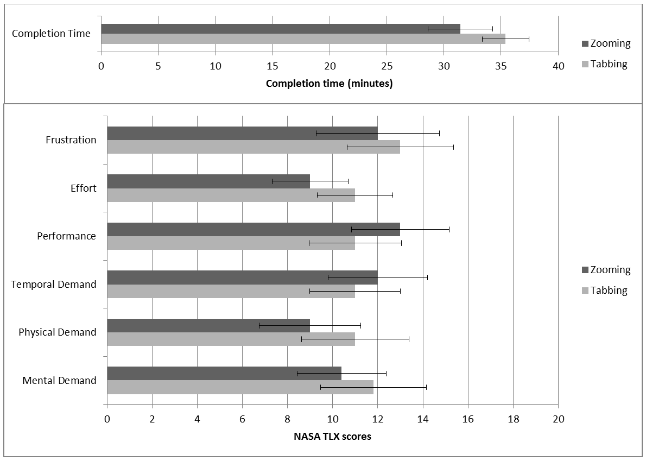


Fig. 6. Time and NASA TLX results.

Using zooming gave users the ability to view the whole form rather than hopping through tabs. In our controlled user tests with users of lower technological background than traditional tablet users, we showed significantly reduced input time and perceived effort with pan-and zoom interaction.

Further user studies are now planned to accurately model form filling process using zooming techniques, in particular to assess the use of zoomed out views in assessing form-completion status and use of zoomed out views as external working memory.

Acknowledgment

We would like to thank the participants in both our observational studies and tablet trials.

References

[1] Ballard, A., Shafer, P. & Sherwood, T. “Customization of tab-order functionality in internet applications”, US Patent, no. 7111243, (2006).

[2] Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D. & Furnas, G. “Pad++: a zoomable graphical sketchpad for exploring alternate interface physics”, *Proc. ACM UIST*, (1994).

[3] Buxton, W.A.S., & Baecker, R.M. Human-computer interaction: a multidisciplinary approach, Morgan Kaufmann, (1987).

[4] Dayao, A.B., Rafieyan, K., Lee, P.J., Graveman, R.F., Yacobi, Y., & Addeo, E.J., “SuperForms: A Security-Enhanced Smart Electronic Form Management System”, *Proc. IEEE Global Telecommunications*, vol. 2, (1990).

[5] Igarashi, T. & Hinckley, K. “Speed-dependent automatic zooming for browsing large documents”, *Proc. ACM UIST 2000, (2000)*.

[6] Perlin, K. & Fox, D. “Pad: an alternative approach to the computer interface”, *Proc. ACM SIGGRAPH*, (1993).

[7] Ministry of communication and Information Technology (MCIT). “Egypt ICT Sector Indicators in brief August 2013", (2013).

Enhancing KLM (Keystroke-Level Model) to Fit Touch Screen Mobile Devices

Karim El Batran
Computer and Information Sciences Dept.
University of Strathclyde
Glasgow, Scotland UK
karim.el-batran@strath.ac.uk

Mark Dunlop
Computer and Information Sciences Dept.
University of Strathclyde
Glasgow, Scotland UK
Mark.Dunlop@cis.strath.ac.uk

ABSTRACT

This short paper introduces an enhancement to the Keystroke-Level Model (KLM) by extending it with three new operators to describe interactions on mobile touchscreen devices. Based on Fitts Law we modelled a performance measure estimate equation for each common touch screen interactions. Three prototypes were developed to serve as a test environment in which to validate Fitts equations and estimate the parameters for these interactions. A total of 3090 observations were made with a total of 51 users. While the studies confirmed each interaction fitted well to Fitts Law for most interaction, it was noticed that Fitts Law does not fit well for interactions with an Index of Difficulty exceeding 4 bits, highlighting a possible maximum comfortable stretch. Based on results, the following approximate movement time for KLM are suggested: 70 ms for a short untargeted swipe, 200 ms for a half-screen sized zoom, and 80 ms for an icon pointing from a home position. These results could be used by developers of mobile phone and tablet applications to describe tasks as a sequence of the operators used and predict user interaction times prior to creating prototypes.

AUTHOR KEYWORDS

Quantitative prediction model; KLM (Keystroke-Level Model); GOMS; Fitts Law; touch screen interaction.

ACM CLASSIFICATION KEYWORDS

H.5.2 User Interfaces: Input devices and strategies.

GENERAL TERMS

Design; Measurement.

INTRODUCTION

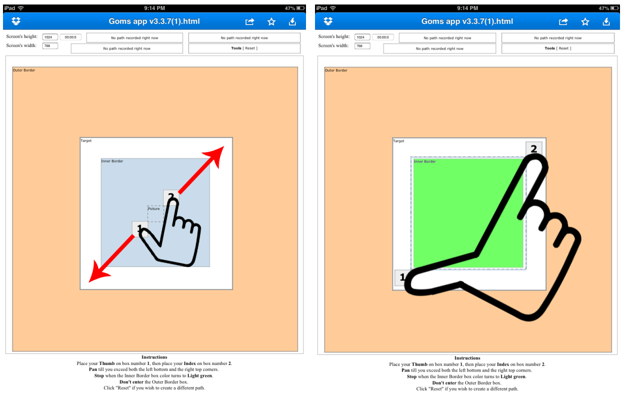
Quantitative prediction models, such as GOMS (Goals, Operators, Methods, and Selection rules) and KLM (Keystroke-Level Model) have been shown to be useful tools in modeling interaction and in deciding between, and filtering out designs (e.g. [1]). KLM is used to estimate the time taken to complete simple data input tasks by combining small timing constants. On physical-key devices it has been widely used to predict task times (e.g. text entry for Korean Language [10], Text Entry Speed on 12-button Phone Keypads [7] etc.).

Many enhancements to KLM have been proposed in the literature to be able to evaluate different techniques. However, there has been little research that improves user behavior modeling techniques to estimate the time taken to achieve common interactions performed on mobile touchscreen devices. In this paper we reported our work on enhancing KLM by extending it with three new operators. In this short paper we report our investigation into modeling three interactions performed on mobile devices and tablets using KLM. Our model is based on suggested times derived from Fitts Law modelling and analysis of how well these interactions fit Fitts Law. We believe this gives an enhancement for developers of smart-phone and tablet applications to predict user interaction times without even needing to create prototypes.

KLM-GOMS MODEL

KLM was introduced as part of the wider GOMS-related work of Card, Moran, and Newell into modelling and quantitatively predicting the skilled and error free performance of users interacting with a text editor [1]. KLM is usually applied in situations that require minimal focussed, scripted interaction with a computer interface or software design.

Confidential submission to MobileHCI 2014.



Instructions
Place your **Thumb** on box number **1**, then place your **Index** on box number **2**.
Pan till you exceed both the left bottom and the right top corners.
Stop when the Inner Border box color turns to **Light green**.
Don't enter the Outer Border box.
Click "Reset" if you wish to create a different path.

Instructions
Place your **Thumb** on box number **1**, then place your **Index** on box number **2**.
Pan till you exceed both the left bottom and the right top corners.
Stop when the Inner Border box color turns to **Light green**.
Don't enter the Outer Border box.
Click "Reset" if you wish to create a different path.

Figure 1 Zoom usability test scenario

Assuming error-free expert user interaction, KLM proved to yield precise estimated results, which is a drawback of similar models [10]. Moreover, when the estimated experimental studies results were considerably different observed values, the estimated difference between two examined designs still proved to be a strong basis for making design choices.

KLM FOR PHONE USERS

Dunlop and Crossan [2] used KLM operator sequences to compare three different text entry methods for traditional physical-key phones (multitap, predictive, and word completion). How and Kan [5] defined 13 operators that directly map onto the phone keyboard interface according to the different input methods. They used videotaped sessions with a small set of subjects and a message-typing task in order to gather new times for these operators. Mori et al. [9] studied how the time values of the original KLM operators apply to mobile phone menu navigation and concluded that the operator values fit well and suggested only minor modifications.

KLM FOR MOBILE DEVICE INTERACTIONS

Until recently the main use of mobile phones was making phone calls, sending text messages, and sometimes basic calendar tasks. Phones used very small displays and small physical buttons as the prime interface. With modern smartphones these buttons have been replaced with larger touch screens. Moreover, other uses are becoming more popular such as taking pictures, surfing the web, social networking, and playing music, videos and games. This adds several new interaction styles that have not yet been treated by interaction models.

There has been surprisingly little published research that includes new interaction techniques such as swipe, zoom, or tabbing, for mobiles and tablets. Luo and John followed-up by Teo and John showed that the method could be soundly applied to handheld devices using stylus-based interfaces [6, 11]. They also presented a tool to automatically generate KLM models from storyboard descriptions and stated that they aimed to apply such research to novel interfaces such as speech and gestures. Holleis et al. adopted and defined a set of operators to give study-based estimates of performance measures. They assumed that developers of mobile applications could then describe tasks as a sequence of the operators they added and predict user interaction times without needing to create prototypes [4]. Closest to our work, as part of their research on user performance of multi-touch gestures on mobile devices, Tran et al. conducted an exploratory study of pinch and spread gestures on smartphones and tablets [12]. Their spread gesture on tablet is equivalent to our Zoom interaction.

CONCEPT AND METHODOLOGY OF PROTOTYPES

The development of three prototypes came in the process of enhancing KLM. These prototypes reflected three common interactions used widely on tablets and mobile devices:

Figure 2. Swipe (left), Tap (right), & Zoom prototypes.

1. Swipe interaction, which enables the user to do different swipe gestures (short or long) to achieve certain tasks (e.g. scrolling through a document etc.).
2. Tap interaction, which estimates the user's tapping response time in comparison to his/her finger travel distance from a home position (e.g. to estimate typing speed or pointing to a target etc.).
3. Zoom interaction, which is used to cause text or other graphics in a window or frame to appear larger on the screen (e.g. zooming, panning a map/photo etc.).

The three prototypes were built to estimate the time taken to conduct these interactions. Fitts Law [3] as a model is used to analyze the general case of time taken estimation for these interactions. It is primarily used in human–computer interaction and ergonomics that predicts the time required to rapidly move to a target area, it is a function of the distance to the target and the size of the target. Of various formulations, the Shannon formulation is widely used [9] and is defined as:

$$T = a + b * ID, \text{ where } ID = log_2\left(\frac{D}{W} + 1\right).$$

Where

- T is the average time taken to complete the movement. It may also be referred to as MT (mean movement time),
- a represents the incorporating reaction time and/or the time required to execute the operator,
- b represents for the inherent speed of the device,
- *D is the distance from start to the center of the target and*
- *W is the width of the target.*

Figure 3 Swipe usability test scenario

Figure 4 Tap usability test scenario

The constant "1" in the formula made a difference from Fitts's original form, especially for low values of the ratio D/W, given the advantage that the ID is always non-negative.Here we are investigating the ability to model the finger touch input for different target sizes and movement distances.

DEVELOPING AND TESTING THE PROTOTYPES

The prototypes were developed in HTML5 / JavaScript and displayed on an iPad Mini (a 7" multi-touch tablet), as shown in figure 2. Three usability studies were conducted to extract completion times (in seconds), for each interaction we tested a range of target size and distances from the starting point to the center of the target (in pixels).

PARTICIPANTS

30 volunteers took part in the Swipe and Zoom usability tests. The participants were aged 22-39, 35% female / 65% male, bachelor graduates, and 70% were familiar with touch screens and 85% were PC users. 21 users conducted the Tap tests, with a similar demographic profile.

TASKS

For Swipe task, forty eight successive attempts were made. As shown in figure 3, users dragged down a box placed on the top center towards the destination box placed on the bottom center of a web page using the participant's index finger. The participant stopped and removed his/her index finger from the screen when the destination box color turned to light green. This was recorded as a successive attempt.

The Tap task, ten successive attempts were made. As shown in figure 4, users were asked to place their index finger on the "Home Position" box near the page's bottom. The participant searched for the "Running Target" box and placed his/her index finger on this target box. He/she removed his/her index finger from the "Running Target" box when its color turned to light green. Afterwards, the participant placed his/her index finger back on the "Home Position" box. This was recorded as a successive attempt.

For Zoom task, twenty six successive attempts were made. As shown in figure 1, each by placing the thumb finger of the participant on the box number 1 and his/her index finger on the box number 2. The participant zoomed till he/she exceeded both the left bottom and the right top corners of the border box, within the target box. He/she stopped and removed his/her thumb and index fingers from the screen when the border box color turned to light green. This was recorded as a successive attempt.

RESULTS

For Swipe task, as shown in figure 5, target width and target amplitude varied across four levels resulting in IDs of 1 to 4 bits. The target width varied from 50 to 400 pixels, while target amplitude varied from 100 to 800 pixels. Mean movement time ranged from 20 ms to 1610 ms with each score derived from over 1440 observations. While the Tap task, the target width varied from 50 to 250 pixels, while target amplitude varied from 125 to 900 pixels. Mean movement time ranged from 50 ms to 470 ms with each score derived from over 210 observations. For the Zoom task, the target width varied from 50 to 150 pixels, while target amplitude varied from 100 to 550 pixels (fixed step of 50 pixels). Mean movement time ranged from 60 ms to 1230 ms with each score derived from over 1440 observations. Regressing MT on ID yields the following prediction equation for movement time (ms):

Swipe interaction: MT = 9.46 + 55.83 ID.

Tap interaction: MT = 52.12 + 14.62 ID.

Zoom interaction: MT = 114.86 ID - 20.45.

By comparing the results with the previous work of Tran et al. [12], both outcomes are largely in line with slightly different prediction equation and higher correlating MT with ID (R^2 Zoom > 0.884).

POOR FIT FOR ID=4

It was noticed in the Swipe and Zoom usability tests that the 4th ID level does not fit in the straight line, which passes by the other 3 levels of ID. Moreover, the constant "a", representing the incorporating reaction time and/or the time required executing the operator, in the prediction equation for movement time of the Zoom usability test is negative. By excluding ID=4, as shown in figure 5, regressing MT on ID yields the following predictions (ms):

- Swipe interaction: MT = 44.78 + 34.64 ID.

Figure 5 Movement Time vs. Index of difficulty for Swipe, Tap, and Zoom tests with IDs of 1 to 3 and 1 to 4 bits.

Interaction	Time
Short untargeted swipe, ½ width of 5" screen	70 ms
½ screen zoom, 100*100 to 350*350 px image	200 ms
Icon pointing of size 200*200 px at a distance of 700 px from home position	80 ms

Table 1. Suggested approximate movement time for KLM

Interaction	R^2
Swipe usability test with IDs of 1 to 3	0.995 ($p<0.05$)
Swipe usability test with IDs of 1 to 4	0.911 ($p<0.1$)
Tap usability test with IDs of 1 to 3	0.972 ($p<0.05$)
Tap usability test with IDs of 1 to 4	0.985 ($p<0.05$)
Zoom usability test with IDs of 1 to 3	0.972 ($p<0.05$)
Zoom usability test with IDs of 1 to 4	0.971 ($p<0.05$)

Table 2. R2 values for interaction execution times.

- Tap interaction: MT = 54.38 + 13.27 ID.
- Zoom interaction: MT = 13.75 + 94.33 ID.

The correlating MT with ID for the Swipe test yields R^2 = 0.995 ($p<0.01$) instead of R^2 = 0.911 ($p<0.1$) (see table 2), which gives a better-fit and more precise prediction.

We believe that the poor fit of ID=4 is indicative that increasing in the spread distance required for a Zoom task of this size is too large for comfort, where the target amplitude is between 500px and 550px (about 98 mm) while the target width is 50px. This result parallels what Hoggan et al. found for pinch, where large pinch distance of over 90mm took significantly longer time to complete tasks and produced an increase in ergonomic failure rate [13].

CONCLUSION

The objective of this paper is to enhance KLM, as a quantitative prediction model, by extending it with new operators necessary to describe interactions used on mobile devices and tablets. It was determined, after conducting three usability tests that estimating the movement time for certain interactions is done using Fitts Law. Furthermore, three equations were formed to estimate the movement time for three interactions; Swipe, Tap, and Zoom. The ID is the variable used to be submitted in any of the three equations in order to calculate the movement time. Based on results, the following approximate movement time for KLM are suggested as shown; a short untargeted swipe will take approximately 70 ms, a half-screen sized zoom will take approximately 200 ms, and an icon pointing from a home position will take approximately 80 ms. We also identified a poor-fit for some high ID operations indicating a possible maximum comfort limit for these tasks.

ACKNOWLEDGMENTS

We are very grateful to our study participants.

REFERENCES

1. Card, S. K., Moran, T. P. and Newell, A. (1983): *The Psychology of Human–Computer Interaction*. Erlbaum.
2. Dunlop, M.D. and Crossan, A. (2000). Predictive Text Entry Methods for Mobile Phones. *Personal Technologies*. 41. 134-143.
3. Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. Experimental Psychology*. 47(6). 381-391.
4. Holleis, P., Otto, F., Hussmann, H., and Schmidt, A. (2007). Keystroke-level model for advanced mobile phone interaction. *Proc. CHI 2007*. 1505-1514.
5. How, Y. and Kan, M. Y. (2005). Optimizing Predictive Text Entry for Short Message Service on Mobile Phones. *Proc. HCI International 2005*.
6. Luo, L. and John, B. E. (2005). Predicting Task Execution Time on Handheld Devices Using the Keystroke-Level Model. *Proc CHI 2005*. 1605-1608.
7. Pavlovych, A., Stuerzlinger, W. (2004). Model for Non-expert Text Entry Speed on 12-button Phone Keypads. *Proc. CHI 2004*. 351-358.
8. MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human–computer interaction. *Human–Computer Interaction*. 7. 91–139.
9. Mori, R., Matsunobe, T., and Yamaoka, T. (2003). A Task Operation Prediction Time Computation Based on GOMS-KLM Improved for the Cellular Phone and the Verification of that Validity. *Proc. ADC 2003*.
10. Myung, R. (2004). Keystroke-level analysis of Korean text entry methods on mobile phones. *Int. J. Human-Computer Studies*. 60(5), 545-563.
11. Teo, L. and John, B. E. (2006). Comparisons of Keystroke-Level Model Predictions to Observed Data. *Extended Abstracts of CHI 2006*. 1421-1426.
12. Tran, J. J., Trewin, S., Swart, C., John, B. E. and Thomas, J. C. (2013). Exploring pinch and spread gestures on mobile devices. *MobileHCI 2013*. 151-16.
13. Hoggan, E., Nacenta, M., Kristensson, P. O., Williamson, J., Oulasvirta, A., and Lehtiö, A. (2013). Multi-touch pinch gestures: performance and ergonomics. Proc. ACM 2013. 219- 222.

Appendix C: Questionnaire

Participant number:
Task:
Date:

Mental Demand:
How mentally demanding was the task?

Very Low Mental Demand Very High Mental Demand

Physical Demand:
How physically demanding was the task?

Very Low Physical Demand Very High Physical Demand

Temporal Demand:
How hurried or rushed was the pace of the task?

Very Low Temporal Demand Very High Temporal Demand

Performance:
How successful were you in accomplishing what you were asked to do?

Perfect performance Failure performance

Effort:
How hard did you have to work to accomplish your level of performance?

Very Low effort Very High effort

Frustration:
How insecure, discouraged, irritated, stressed, and annoyed were you?

Very Low frustration Very High frustration

رقم المشترك :

المهمة :

تاريخ :

المطلب الذهني :
كيف تطلبة هذه المهمة جهدا ذهنيا ؟

قليلا جدا   عاليا جدا

المطلب البدني :
كيف تطلبة هذه المهمة جهدا بدنيا ؟

قليلا جدا   عاليا جدا

المطلب الزماني :
كيف كانت وتيرة هذه المهمة ؟

بطئ جدا   سريعة جدا

الأداء :
مدى النجاح في إنجاز ما طلب منك أن تفعل ؟

إخفاق   اتمام

الجهد :
كيف كانت الصعوبه للعمل على تحقيق هذا المستوى من الأداء ؟

قليلا جدا   عاليا جدا

الإحباط :
ما مدي شعورك بعدم الراحة ، بالإحباط ، غضب و الانزعاج ؟

قليلا جدا   عاليا جدا

Appendix D: Translated Arabic Phrases

No.	English Sentence	Arabic Sentence
1	The proprietor was unavailable.	متوفر غير المالك كان
2	Would you like to come to my house.	بيتي إلى تأتي أن تريد هل
3	Can I skate with sister today.	اليوم شقيقته مع تزلج أستطيع
4	This camera takes nice photographs.	لطيفة صورا تلتقط الكاميرا
5	Dolphins leap high out of the water.	الماء من عاليا قفزة الدلافين
6	This leather jacket is too warm.	جدا دافئة الجلدية السترة هذه
7	There are winners and losers.	وخاسرون رابحون هناك
8	Bring the offenders to justice.	للعدالة الجناة تقديم
9	Questioning the wisdom of the courts.	المحكمه الحكمة في التشكيك
10	The punishment should fit the crime.	الجريمة مع تتناسب ان للعقوبة ينبغي
11	Watch out for low flying objects.	الطائرة الأجسام انخفاض من احترس
12	We must redouble our efforts.	جهودنا نضاعف أن علينا يجب
13	Great disturbance in the force.	القوة في كبير اضطراب
14	A feeling of complete exasperation.	الكامل الغضب شعور
15	Win first prize in the contest.	المسابقة في الأولى بالجائزة الفوز
16	There will be some fog tonight.	الليلة الضباب بعض هناك يكون وسوف
17	Soon we will return from the city.	المدينة من نعود سوف قريبا
18	A glance in the right direction.	الصحيح الاتجاه في لمحة
19	You should visit to a doctor.	طبيب زيارة عليك يجب
20	A good stimulus deserves a good response.	جيدة استجابة يستحق جيد حافز
21	The world is a stage.	المسرح هو العالم
22	A most ridiculous thing.	سخيف شيء
23	You will loose your voice.	صوتك تفقد سوف
24	Parking tickets can be challenged.	السيارات وقوف تذاكر علي التغلب يمكن
25	We run the risk of failure.	الفشل لخطر نتعرض نحن
26	The biggest hamburger I have ever seen.	حياتي في رأيته همبرغر أكبر
27	Give me one spoonful of coffee.	القهوة من ملعقة اعطيني
28	Rejection letters are discouraging.	مشجعة غير الرفض خطابات
29	Everybody looses in custody battles.	الحضانة معارك في الجميع يفقد
30	Only an idiot would lie in court.	المحكمة في يكذب الذي الاحمق فقط
31	The generation gap gets wider.	أوسع اصبح الأجيال بين الفجوة
32	Physics and chemistry are hard.	صعبه مةار والكيمياء الفيزياء
33	The dog will bite you.	الكلب يعضك وسوف
34	A psychiatrist will help you.	يساعدك سوف نفسي طبيب
35	Just what the doctor ordered.	الطبيب أمر ما مجرد
36	The king sends you to the tower.	البرج الي يرسلك الملك
37	Fine but only in moderation.	الاعتدال في فقط ولكن جيد
38	The library is closed today.	اليوم مغلاقه المكتبة
39	Everyone wants to win the lottery.	اليانصيب في الفوز يريد الجميع

No.	English Sentence	Arabic Sentence
40	Completely sold out of that.	بالكامل بيعت
41	That land is owned by the government.	الحكومة الي الأراضي هذه ملكية وتعود
42	The daring young man.	الجرئ الشاب هذا
43	February has an extra day.	اضافيا يوما به فبراير
44	Teaching services will help.	التدريس خدمات تساعد وسوف
45	The fire raged for an entire month.	كامل شهر لمدة النار اندلعت
46	Time to go shopping.	للتسوق للذهاب الوقت حان
47	This mission statement is baloney.	Unable to translate
48	The picket line gives me the chills.	
49	Starlight and dewdrop.	
50	Buckle up for safety.	