

INPUT EVALUATION OF AN EYE-GAZE-GUIDED INTERFACE:

KALMAN FILTER VS. VELOCITY THRESHOLD

EYE MOVEMENT IDENTIFICATION

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree

Master of SCIENCE

by

Sandeep A. Munikrishne Gowda, B.E.

San Marcos, Texas

December 2009

ACKNOWLEDGEMENTS

I am grateful to my thesis advisor Oleg Komogortsev who inspired me during my initial year of graduate studies to take up the thesis option. It is an honor for me to have him as my thesis advisor, who constantly provided me with immense guidance and support throughout my thesis work.

Special thanks to my buddy Dohyong Koh for his work on the Kalman filter and his support during my thesis work.

My sincere thanks to all the class members of the Advanced HCI, Fall 2008 who worked on the initial design and development of the iGaze application.

Finally, I would like to thank the members of my thesis committee, my parents and my friends who have supported me during the graduate studies.

This manuscript was submitted on November 11, 2009.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF FIGURES	vi
ABSTRACT	vii
CHAPTER	
I. INTRODUCTION.....	1
II. REAL-TIME EYE MOVEMENT CLASSIFICATION	4
Velocity Threshold Identification (I-VT)	4
Kalman Filter Classification (I-KF).....	8
Predict	9
Update	9
Saccade Detection by Chi Square Test	11
III. INPUT EVALUATION	15
Accuracy test.....	15
ISO 9241-9 standard	16
Eye gaze-guided system design	16
Layout	16
Individual component size	16
Visual Feedback.....	17
IV. iGAZE INTERFACE	18
Modes of interaction	19
Album mode.....	19
Picture mode	21

V. METHODOLOGY	25
Participants.....	25
Apparatus	25
Procedure	25
VI. RESULTS.....	28
Accuracy Test	28
Tasks Completed & Completion Time	28
Completion Time vs. Accuracy	29
Completion Time vs. Data Loss.....	32
Questionnaire	35
VII. DISCUSSION	37
VIII.CONCLUSION.....	38
APPENDIX A	40
APPENDIX B	50
BIBLIOGRAPHY	55

LIST OF FIGURES

Figure	Page
1. Pseudo code for the I-VT model.	6
2. Control flow for the I-VT model	7
3. Pseudo code for I-KF model.	13
4. Control flow for the I-KF model.....	14
5. iGaze – album mode.	20
6. iGaze – picture mode.	22
7. Visual feedback provided by gidgets.	24
8. Completion time vs. Accuracy for I-VT	30
9. Completion time vs. Accuracy for I-KF	31
10. Completion Time vs. Data Loss for I-VT	33
11. Completion Time vs. Data Loss for the I-KF	34
12. Device Assessment Questionnaire results.	36

ABSTRACT

INPUT EVALUATION OF AN EYE-GAZE-GUIDED INTERFACE:

KALMAN FILTER VS. VELOCITY THRESHOLD

EYE MOVEMENT IDENTIFICATION

by

Sandeep A. Munikrishne Gowda, B.E.

Texas State University-San Marcos

December 2009

SUPERVISING PROFESSOR: OLEG V. KOMOGORTSEV

This thesis evaluates the input performance capabilities of Velocity Threshold (I-VT) and Kalman Filter (I-KF) eye movement detection models when employed for eye-gaze-guided interface control. I-VT is a common eye movement identification model employed by the eye tracking community, but it is neither robust nor capable of handling high levels of noise present in the eye position data. Previous research implies that use of a Kalman filter reduces the noise in the eye movement signal and predicts the signal

during brief eye movement failures, but the actual performance of I-KF was never evaluated. We evaluated the performance of I-VT and I-KF models using guidelines for ISO 9241 Part 9 standard, which is designed for evaluation of non keyboard/mouse input devices with emphasis on performance, comfort, and effort. Two applications were implemented for the experiment: 1) an accuracy test 2) a photo viewing application specifically designed for eye-gaze-guided control. Twenty-one subjects participated in the evaluation of both models completing a series of tasks. The results indicates that I-KF allowed participants to complete more tasks with shorter completion time while providing higher general comfort, accuracy and operation speeds with an easier target selection than the I-VT model. We feel that these results are especially important to the engineers of new assistive technologies and interfaces that employ eye-tracking technology in their design.

CHAPTER I

INTRODUCTION

An eye-tracking device can be used as an interactive input modality for users with disabilities or as an additional interaction channel for normal users (Douglas, Kirkpatrick, & MacKenzie, 1999). Engineering challenges behind using this type of input are 1) eye-tracking failures due to eye squinting, eye moisture, and blinks 2) noise due to eye-tracking hardware inaccuracies and micro eye movements. These challenges can be solved by processing raw eye position signals and classifying them into meaningful components such as fixations (eye movements that occur when gaze is dwelling on objects), saccades (eye movements between two separate fixations), and pursuits (eye movements that occur when eyes are tracking moving objects). Fixations (dwell-time) are the most common modality for an eye-gaze-guided computer interaction (Kumar, Paepcke, & Winograd, 2007; Sibert & Jacob, 2000; Zhai, Morimoto, & Ihde, 1999). This modality of interaction assumes that a fixated object is selected when the duration of a fixation reaches a predefined threshold. Very few works exist that employ saccade based interaction (Urbina & Huckauf, 2007). To the best of our knowledge pursuit-based interaction is an unexplored topic in the Human Computer Interaction (HCI) community. In this thesis we consider only fixation-based mode of interaction.

Several models exist for eye movement classification, including the most used Velocity-based Threshold (I-VT) model described by Salvucci and Goldberg (2000). The

I-VT model is usually selected because of the ease of its implementation and its low computational cost, but the model is not robust and is not capable of handling high levels of noise present in the eye position data.

The Kalman filter is a recursive estimator that computes a future estimate of the dynamic system state from a series of incomplete and noisy measurements. Eye trackers frequently fail to report eye position data and the reported data are susceptible to noise due to the individual anatomical properties of users and limited spatial resolution of the equipment. Therefore, Kalman filter framework can be applied to process raw eye position data to provide more accurate and robust estimations of the eye position signal. At the same time, the Kalman filter is capable of classifying eye movements (Sauter, Martin, Di Renzo, & Vomscheid, 1991). The use of a Kalman filter (I-KF) in a real-time eye-gaze-guided computer interface was first discussed by Komogortsev and Khan (2007) where the research indicated that the filter can be successfully used during eye-tracking failures. Kumar et al. (2008) presented the case where a Kalman filter provided smoothing to a raw eye position signal, thereby increasing the stability of the input. Unfortunately, both research groups did not provide a comprehensive evaluation of the interface performance driven by the Kalman filter.

This thesis provides an objective and subjective performance analysis of I-KF and I-VT models, by employing an accuracy test and a standalone real-time eye-gaze-driven photo viewing application (iGaze).

Performance of I-VT and I-KF was conducted following several evaluation guidelines of the ISO 9241 Part 9 standard. This standard allows conducting an evaluation of non keyboard/mouse based input devices. Previously, Zhang and

MacKenzie (2007) successfully applied the ISO 9241 Part 9 standard to the eye-gaze guided task for a simple target selection. In our research, we applied the standard guidelines to a full standalone eye-gaze-guided application.

Twenty one subjects participated in the evaluation where eye movement identification was done by either I-VT or I-KF model. In addition to the accuracy test, each subject had to complete a series of five tasks using the iGaze application. The results allowed us to conclude that participants with I-KF were able to complete more tasks on average along with a shorter completion time while providing smoother, higher accuracy, higher operation speed, and easier target selection than the I-VT model.

Chapter II describes the I-VT and I-KF real time eye movement classification models along with the pseudo code for each of the models. The input evaluation, an accuracy test, the ISO 9241-9 standard, and the design principles to be followed for developing an eye gaze guided system are described in Chapter III. The design principles outlined in Chapter III are employed in the creation of the iGaze photo viewing application, which is described in detail in Chapter IV. The methodology for this thesis research is described in Chapter V. The results obtained as a part of this research work is explained in Chapter VI and Chapter VII discusses the observations made during this research work. The conclusion and directions for future research work are provided in the Chapter VIII.

CHAPTER II

REAL-TIME EYE MOVEMENT CLASSIFICATION

Many eye movement classification models exist that define ways to identify the different eye movement types. Among these, very few models can be implemented for real time classification. This chapter explains two such real time eye movement classification capable models, I-VT and I-KF and also provides the pseudo code for both of the eye movement identification models.

Velocity Threshold Identification (I-VT)

I-VT distinguishes fixations and saccades based on the observation of velocities between two separate eye positions. If the sampled velocity is greater than the threshold then the corresponding eye position sample is marked to be a part of a saccade; otherwise the eye position sample is marked to be a part of a fixation (Salvucci & Goldberg, 2000). Consecutive eye position points, classified as fixation, are collapsed into a single fixation with a center coordinates computed as a centroid of all points in the fixation. Classified fixations are subsequently merged into larger fixations by the criteria based on two parameters: latency and distance between two subsequent fixation points. Figure 1 presents the pseudo code for the I-VT model. Velocity separation threshold is the main parameter responsible for the correct performance of the I-VT model. Different literature sources indicate different values for this threshold. Salvucci and Goldberg (2000) indicate values above 300°/s for saccades and below 100°/s for fixations. Komogortsev

and Khan (2007) suggest $30^{\circ}/s$ and $5^{\circ}/s$. Practical approach necessitates the empirical selection of such parameter with the value of $75^{\circ}/s$ selected in our system.

Figure 2 shows the control flow of the real time I-VT model. The eye position sample from the eye tracker is provided as an input to the “Pre-processing” module that converts the coordinates of each eye position sample from eye tracker units to degrees. Next the “Velocity” module computes the velocity between the current and previous eye position samples. This velocity, along with the converted eye position data are supplied to the “Identification” module for classifying the eye movement type.

The I-VT model is simple to implement, but the model has low tolerance to noise caused by equipment failures and/or micro eye movements.

Algorithm: I-VT

Input: Current eye position sample, Previous classified eye position sample

Output: Classified eye position sample

Process the current eye position sample

Calculate the velocity between current and previous samples

Movement Identification (Velocity, Threshold)

If the velocity is below the threshold then,

If current and previous eye positions are within the micro saccade amplitude then set the current sample as a Fixation and update the duration of the Fixation.

Otherwise classify current eye position sample as Saccade.

Return classified eye position sample

Figure 1. Pseudo code for the I-VT model.

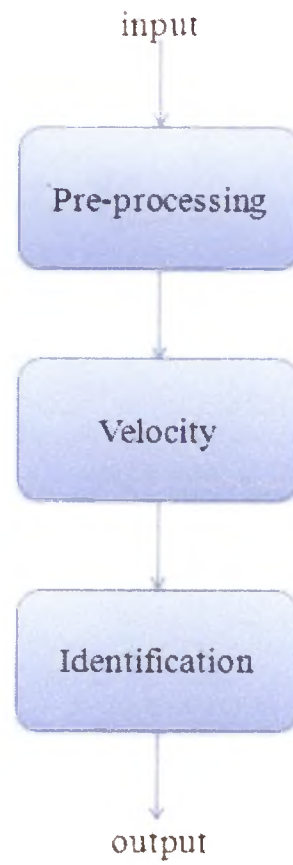


Figure 2. Control flow for the I-VT model. The control flow for the real time I-VT model.

Kalman Filter Classification (I-KF)

The Kalman filter is a recursive estimator that computes a future estimate of the dynamic system state from a series of incomplete and noisy measurements. A Kalman Filter minimizes the error between the estimation of the system's state and the actual system's state. Only the estimated state from the previous time step and the new measurements are needed to compute the new state estimate. Many real dynamic systems do not exactly fit this model; however, because the Kalman filter is designed to operate in the presence of noise, an approximate fit is often adequate for the filter to be quite useful (Brown & Hwang, 1996).

The Kalman Filter addresses the problem of trying to estimate the state $x \in \mathbb{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation (Brown and Hwang, 1997);

$$x_{k+1} = A_{k+1}x_k + B_{k+1}u_{k+1} + w_{k+1} \quad (1)$$

with the measurement

$$z_k = H_k x_k + v_k \quad (2)$$

The n -by- n state transition matrix A_{k+1} relates the state at the previous time step k to the state at the current step $k+1$ in the absence of either a driving function or process noise. B_{k+1} is an n -by- m control input matrix that relates m -by- 1 control vector u_{k+1} to the state x_k . w_k is an n -by- 1 system's noise vector with an n -by- n covariance matrix Q_k . $p(w_k) \sim N(0, Q_k)$. The measurement vector z_k contains state variables that are measured by the instruments. H_k is a j -by- n observation model matrix which maps the state x_k into the measurement vector z_k . v_k is a measurement noise j -by- 1 vector with covariance R_k . $p(v_k) \sim N(0, R_k)$.

The Discrete Kalman filter has two distinct phases that compute the estimate of the next system's state (Brown & Hwang, 1997).

Predict

Predict the state vector ahead:

$$\hat{x}_{k+1}^- = A_{k+1}x_k + B_{k+1}u_{k+1} \quad (3)$$

The \hat{x}_{k+1}^- is a prediction of the position of the future gaze.

Predict the error covariance matrix ahead:

$$P_{k+1}^- = A_{k+1}P_kA_{k+1}^T + Q_{k+1} \quad (4)$$

The predict phase uses the previous state estimate to predict the estimate of the next system's state.

Update

Compute the Kalman gain:

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1} \quad (5)$$

Update the estimate of the state vector with a measurement z_{k+1} :

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1}^-) \quad (6)$$

Update the error covariance matrix:

$$P_{k+1} = (I - K_{k+1}H_{k+1})P_{k+1}^- \quad (7)$$

Two State Kalman Filter (TSKF) models an eye as a system with two states: position and velocity. The acceleration of the eye is modeled as white noise with known maximum acceleration and used in the design of the Q_k matrix for the Equation (4).

The TSKF models an eye as a system which has two state vectors x_k and y_k .

$$x_k = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (8)$$

where $x_1(k)$ is horizontal coordinate of the gaze position and $x_2(k)$ is horizontal eye-velocity at time k .

$$y_k = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \quad (9)$$

where $y_1(k)$ is vertical gaze position and $y_2(k)$ is vertical eye-velocity at time k .

The state transition matrix for both horizontal and vertical states is:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (10)$$

where Δt is the eye-tracker's eye-position sampling interval.

The observation model matrix for both state vectors is:

$$H = [1 \quad 0] \quad (11)$$

By definition the covariance matrix for the measurement noise is $R_k = E[(v_k - E(v_k))(v_k - E(v_k))^T]$. Because only eye position is measured v_k is a scalar making $R_k = \text{VAR}[v_k] = \delta_v^2$, where δ_v is the standard deviation of the measurement noise. In this thesis it is assumed that the standard deviation of the measurement noise relates to the accuracy of the eye tracker and is bounded by one degree of the visual angle. Therefore δ_v was conservatively set to 1° . In cases when the eye tracker fails to detect eye position coordinates the standard deviation of measurement noise is assigned to be $\delta_v = 120^\circ$. The value of 120° is chosen empirically, allowing the Kalman Filter to rely more on the predicted eye position coordinate \hat{x}_k^- .

The TSKF is initialized with zero valued initial vectors \hat{x}_0 , \hat{y}_0 and an identity error covariance matrix P_0 .

By definition process noise covariance matrix is $Q_k = E[(w_k - E(w_k))(w_k - E(w_k))^T]$, where w_k a 1x2 system's noise vector is $w_k = [w_1(k) \quad w_2(k)]^T$. TSKF

Simple model assumes that variables $w_1(k)$ are uncorrelated between each other (velocity is independent of eye position), i.e., $E[(w_m(k)w_n(k))] = E[(w_m(k))]E[w_n(k)]$ for all $n \neq m$ and $p(w_1(k)) \sim N(0, \delta_1^2)$, $p(w_2(k)) \sim N(0, \delta_2^2)$. These assumptions generate following system's noise covariance matrix: $Q_k = \begin{bmatrix} \delta_1^2 & 0 \\ 0 & \delta_2^2 \end{bmatrix}$. Simple model assumes that the standard deviation of the eye position noise $w_1(k)$ is connected to the characteristics of the eye fixation movement. Each eye fixation consists of three basic eye-sub-movements: drift, small involuntary saccades and tremor (Yarbus, 1967). Among those three, involuntary saccades have the highest amplitude - about a half degree of the visual angle; therefore δ_1 is set conservatively to 1° . Standard deviation value for eye velocity was selected to be $\delta_2 = 1^\circ/\text{s}$.

Saccade Detection by Chi Square Test

Chi Square Test was originally employed by Sauter et al. (1991) to detect the onset and the offset of a saccade. Chi square test monitors the difference between predicted and observed eye-velocity:

$$\chi^2 = \sum_{i=1}^p \frac{(\dot{\theta}_i^- - \dot{\theta}_i)^2}{\delta^2} \quad (12)$$

where $\dot{\theta}_i^-$ is the predicted eye-velocity computed with Equation (12) and $\dot{\theta}_i$ is the observed eye-velocity. δ is the standard deviation of the measured eye-velocity during the sampling interval under consideration. Once a certain threshold of the χ^2 is achieved a saccade is detected (value of 25 is used in our system). It was reported that the filter behaves better if the standard deviation δ is a constant (Komogortsev & Khan, 2007). For our experiment, we used values $\delta^2 = 1000$ and $p=5$.

The I-KF method is able to distinguish fixations and saccades efficiently in a real time eye gaze guided system. It means that the I-KF can compensate for situations when the part of the eye position data when is missing or invalid.

The pseudo code for the I-KF model is presented by Figure 3. The control flow for the real time I-KF model is illustrated by Figure 4. The eye position sample from the eye tracker is provided as an input to the “Pre-processing” module that converts the coordinates of each eye position sample from eye tracker units to degrees. The module also computes the velocity between the current and previous eye position samples. The converted eye position data is supplied to the “Kalman Filter” module where the predicted velocity for the current eye position is computed. The “Chi-square Test” performed by Equation 12 is applied onto the predicted velocity and computed velocity. The result from the “Chi-square Test” module is employed to classify the initial eye position sample as a fixation or a saccade in the “Identification” module.

Algorithm: I-KF**Input:** Current eye position sample, Previous classified eye position sample**Output:** Classified eye position sample

Process the current eye position sample

Calculate the velocity between current and previous samples

Apply the Kalman Filter

Apply the Chi-square Test

Movement Identification (Threshold)

If the Chi-square test value is below the threshold then,
If current and previous eye positions are within the
micro saccade amplitude then set the current sample as a
Fixation and update the duration of the Fixation.
Otherwise classify current eye position sample as
Saccade.

Return classified eye position sample

Figure 3. Pseudo code for I-KF model.

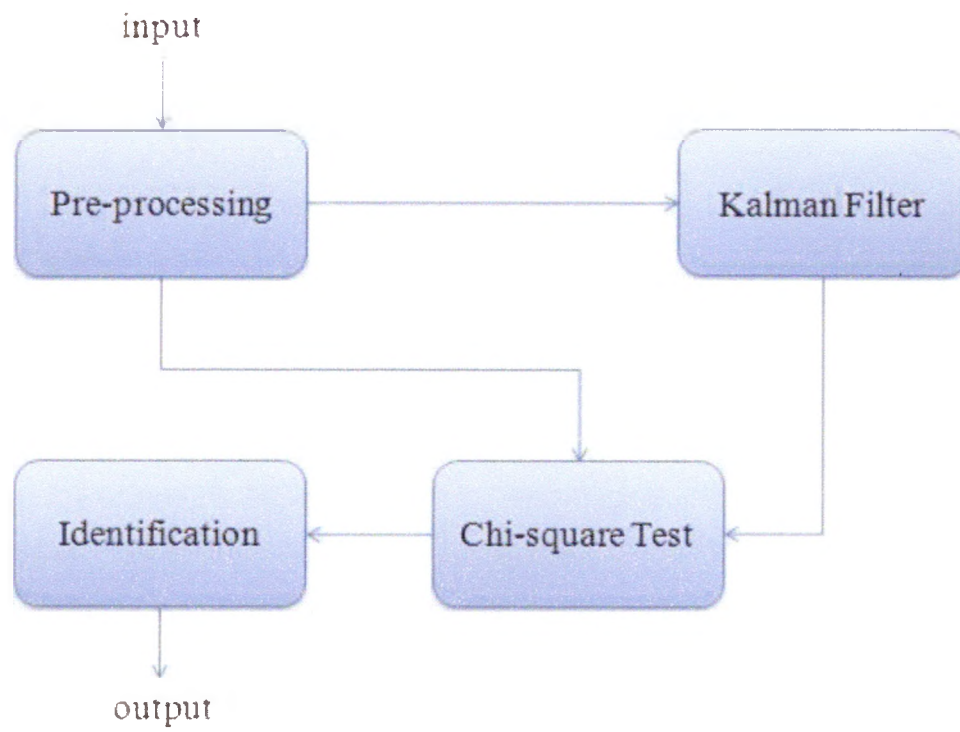


Figure 4. Control flow for the I-KF model. The control flow for the real time I-KF model.

CHAPTER III

INPUT EVALUATION

This chapter explains the two methods of evaluation used to test the input performance capabilities of I-VT and I-KF models. The first method is an accuracy test that provides an objective measurement of accuracy performance. The second method uses an eye-gaze-guided application with performance evaluated through a series of tasks and a questionnaire suggested by the ISO 9241-9 standard. The use of a real application allows testing for objective measurements such as task completion time and the number of tasks completed, and the questionnaire provides a subjective evaluation in terms of performance, comfort, and effort.

Accuracy test

This procedure involves participants looking at 17 sequentially presented points that are uniformly distributed on the computer screen. When a subject fixates at each point, the raw eye position signal is processed by either I-VT or I-KF and corresponding fixation parameters such as location coordinates, the onset time, and the duration are determined. The coordinates of the eye position within the detected fixations are compared to the center of presented stimulus. This allows for the computation of error between reported location of the gaze and the actual gaze point. At the end of the recording, the error values are averaged between all points and presented on the computer screen. Additionally, an accuracy test computes and presents data loss parameter that

indicates the amount of erroneous (not detected) eye position samples provided by an eye tracker for the participant.

ISO 9241-9 standard

The ISO 9241-9 standard (ISO, 2000) is used for the evaluation of computer pointing devices with suggested measurement of performance and comfort. The standard provides a questionnaire for evaluating performance, comfort, and effort of a given input modality.

Eye gaze-guided system design

The design challenge of any eye-gaze-guided computer system can be separated into three categories: layout design, individual component size selection, and visual feedback. We suggest the following guidelines to address the issues presented in each category.

Layout

The minimum spacing between each individual component of the interface should not be less than the reported eye tracker accuracy. Usually the accuracy value is around 0.5° (Duchowski, 2007). The accuracy of the eye-tracker equipment degrades closer to the periphery of the computer screen; therefore, the layout of the individual interface components should provide an increased spacing at the boundary of the computer screen.

Individual component size

The size of the individual component should not be less than reported eye-tracker's accuracy to provide an accurate selection of this component by a fixation detection algorithm. An ideal component size would be around 2° of the visual angle due to the similar size of the human fovea – region in our eye providing the highest acuity of

vision (Irwin, 1992). We recommend keeping individual component size between 0.5° - 2° .

Visual Feedback

Eye-gaze-guided systems do not have a mouse cursor following the eye gaze because of the chasing effect that appears when the reported gaze location does not match with the actual gaze position on the screen (Duchowski, 2007). Nevertheless, the eye-gaze-guided system must indicate to the user that the interface component is currently being attended to. We recommend highlighting the boundary of individual interface component with different colors when user's gaze is directed toward this component in order to indicate the longevity of attention. Users successfully use this mechanism for the interface control and validation.

CHAPTER IV

iGAZE INTERFACE

With the eye gaze system design guidelines outlined, we designed an iGaze interface as an image viewing application capable of navigating through a set of grouped pictures (albums) using eye movements as the primary input. This chapter provides the details about the layout design and the functionality of the iGaze application.

The iGaze provides functionality of scrolling through a list of albums, selecting and expanding the desired album into a viewable set of pictures, and finally enlarging the selected picture providing a full screen view to the user. Component selection was done by dwell time selection method, where the selection occurs based on the specified fixation duration (200 ms.). As reported by Tien and Atkins (2008) the value of 200 ms. provides an adequate balance between speed of interaction and accuracy.

The iGaze application was designed according to the eye-gaze-guided system design, i.e., all individual components called gidgets (term indicating the specificity of widgets designed for eye-gaze selection.) had spacing greater than 0.5° with individual gidget size larger than 1° . The size of gidgets was further increased at the screen boundaries to compensate for hardware inaccuracies.

Modes of interaction

Album mode

In this mode, the interface displays list of albums on the left side and grid of thumbnails of the currently selected albums on the right side. Also, both the album list and the thumbnails can be scrolled up and down using the “UP” and “DOWN” buttons. Figure 5 shows a screenshot of the album mode of the iGaze.

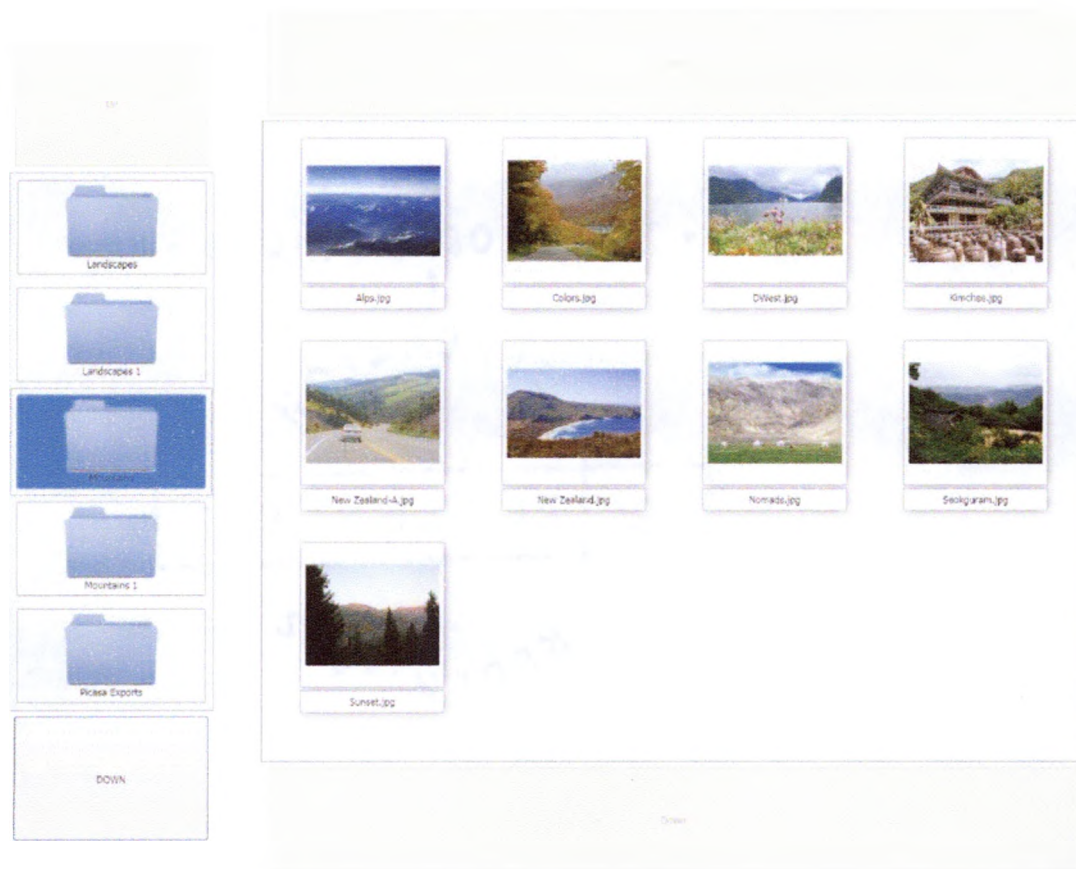


Figure 5. iGaze – album mode.

Picture mode

In this mode, the thumbnail that gets selected in the album mode is displayed with enlarged view at the top right corner on a screen. At the bottom, thumbnails of other pictures in the album are listed horizontally across the screen with two buttons at each corner that enable scrolling left or right. Located on the left is a button that enables to switch back to album mode. Figure 6 shows a screenshot of the picture mode of the iGaze.



Figure 6. iGaze – picture mode.

Visual feedback is provided for every gadget when an eye-gaze is directed toward the component. Specifically, all gadgets have a border that starts to glow blue as soon as the eye-gaze enters the gadget area. The border's glow color turns from blue to red as the fixation duration increases. The red border glow is designed to provide an indication that the gadget is about to get selected. Figure 7 shows the screenshots that indicate the visual feedback for the iGaze.



Figure 7. Visual feedback provided by gadgets.

CHAPTER V

METHODOLOGY

This chapter outlines the test procedure and the apparatus used for evaluating the input performance through the accuracy test and the iGaze application.

Participants

A total of 21 participants volunteered for the evaluation test. Participants' ages were from 18 to 35 (*mean* = 22.3). None of the participants had prior experience with eye tracking. Among these participants, 11 had normal vision and 10 wore glasses or contacts.

Apparatus

The experiments were conducted with Tobii x120 eye tracker, which is represented by a standalone unit connected to a 19 inch flat panel screen with resolution of 1280x1024. The eye tracker performs binocular tracking with the following characteristics: accuracy 0.5°, spatial resolution 0.2°, drift 0.3° with eye position sampling frequency of 120Hz. Tobii x120 model allows 300x220x300 mm freedom of the head movement. Nevertheless, a chin rest was employed for higher accuracy and stability.

Procedure

An accuracy test was applied to every participant at the beginning of each experiment. Next, each participant was asked to complete a sequence of five tasks using

an iGaze application. Prior to each task, a subject was presented with an image cropped from one of the pictures stored in the iGaze application. The goal of each task was to find the original picture within three minutes. After each task, completion time was recorded. If the participant failed to find the picture, we marked it as time out. Half of the subjects were assigned to complete the tasks using the I-KF model and the remaining half were assigned to complete the tasks using the I-VT model. Such subject assignment was employed to negate possible interface learning effects that would occur if a subject would perform a series of tasks using both models sequentially.

At the end of the procedure, subjects were asked to complete a survey containing a Device Assessment Questionnaire suggested by ISO 9241-9 guidelines, with some questions modified to be more task related. Table 1 provides the list of the Device Assessment Questionnaire. All of the questions were rated on a 5-point Likert scale.

Table 1. Device Assessment Questionnaire	
Sl. No.	Question
1.	Smoothness during operation was(very rough to very smooth)
2.	The mental effort required for operation was (too low to too high)
3.	The physical effort required for operation was (too low to too high)
4.	Accurate pointing was (easy to difficult)
5.	Operation speed was(too fast to too slow)
6.	Eye fatigue (none to very high)
7.	Target Selection(easy to very difficult)
8.	Which will you prefer the eye gaze interface or interface with keyboard
9.	Shoulder fatigue (none to very high)
10.	Neck fatigue (none to very high)
11.	General comfort (very uncomfortable to very comfortable)
12.	Overall, the interface was (very difficult to use to very easy to use)

CHAPTER VI

RESULTS

The results obtained after conducting the accuracy test and the sequence of the tasks using iGaze are described in this chapter.

Accuracy Test

The average fixation position detection error was 0.14° ($SD=0.011$) for I-VT and 0.126° ($SD=0.007$) for I-KF, indicating an improvement of accuracy of approximately 10% achieved by employing the I-KF model. The result was statistically significant ($F(1,35067)=168.86, p<0.001$).

Tasks Completed & Completion Time

The average number of tasks completed was 3.22 for I-VT ($SD=1.48$) and 3.8 ($SD=1.03$) for I-KF, indicating that participants were able to complete approximately 18% more tasks using I-KF model. The result was statistically significant ($F(1,17)=7.15, p<0.01$).

The average completion time was 92.59 seconds ($SD=39.26$) for I-VT and 88.06 seconds ($SD=18.05$) in I-KF indicating that participants were able to complete tasks on average 5% faster using I-KF. The result was statistically significant ($F(1,17)=4.86, p<0.04$).

Completion Time vs. Accuracy

Figure 8 and Figure 9 are the plots that illustrate the task completion times given the specific accuracy error for I-VT and I-KF models respectively. The accuracy error indicates the difference between the actual eye-gaze position and position reported by the eye-tracker. Larger accuracy errors result in erroneous or unsuccessful selections, therefore hindering interaction performance.

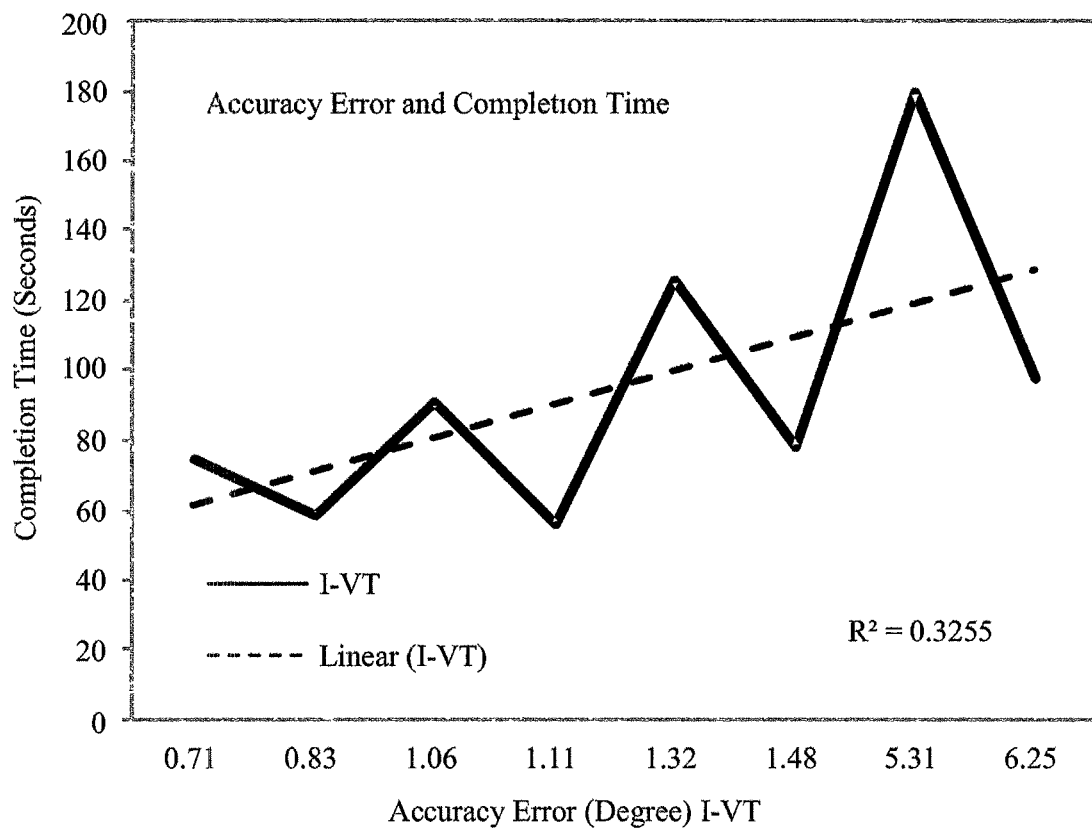


Figure 8. Completion time vs. Accuracy for I-VT. Represents recorded range for the I-VT model.

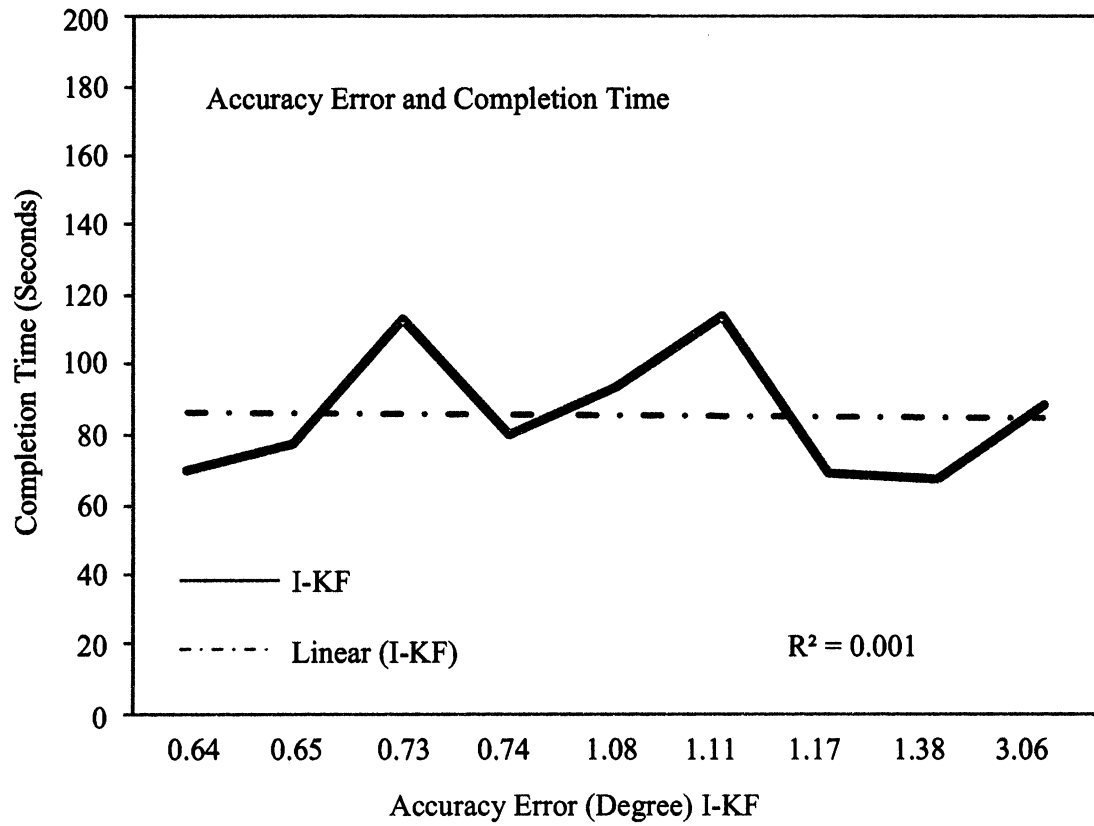


Figure 9. Completion time vs. Accuracy for I-KF. Represents recorded range for the I-KF model.

The results indicate that lower accuracy provided a higher negative impact for the I-VT model, increasing the amount of time required to complete a task. It is apparent that completion time increased with lower accuracy value. In terms of the I-KF model, the accuracy error did not have such a significant effect, *e.g.*, completion time at 0.71° accuracy was approximately the same as at 3° of accuracy.

Completion Time vs. Data Loss

Figure 10 and Figure 11 are plots that show the comparison between the completion time and the data loss for I-VT and I-KF models respectively. The data loss indicates the quantity of invalid data from the eye tracker for each subject. Also, the data loss was computed as a part of the accuracy test. Data loss can have a substantial negative effect on the eye-gaze-guided system performance causing delays and incorrect fixation detections. As a result, it is important to evaluate the performance of the eye-gaze-guided system given the specific level of data loss.

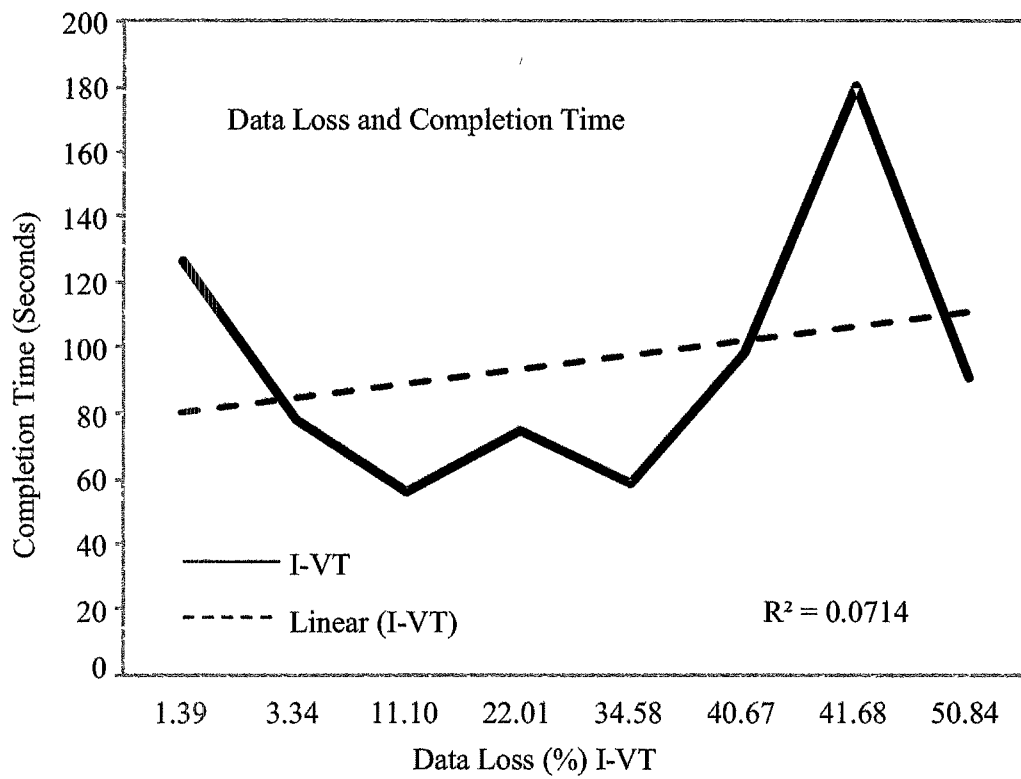


Figure 10. Completion Time vs. Data Loss for I-VT. Represents recorded range for the I-VT model.

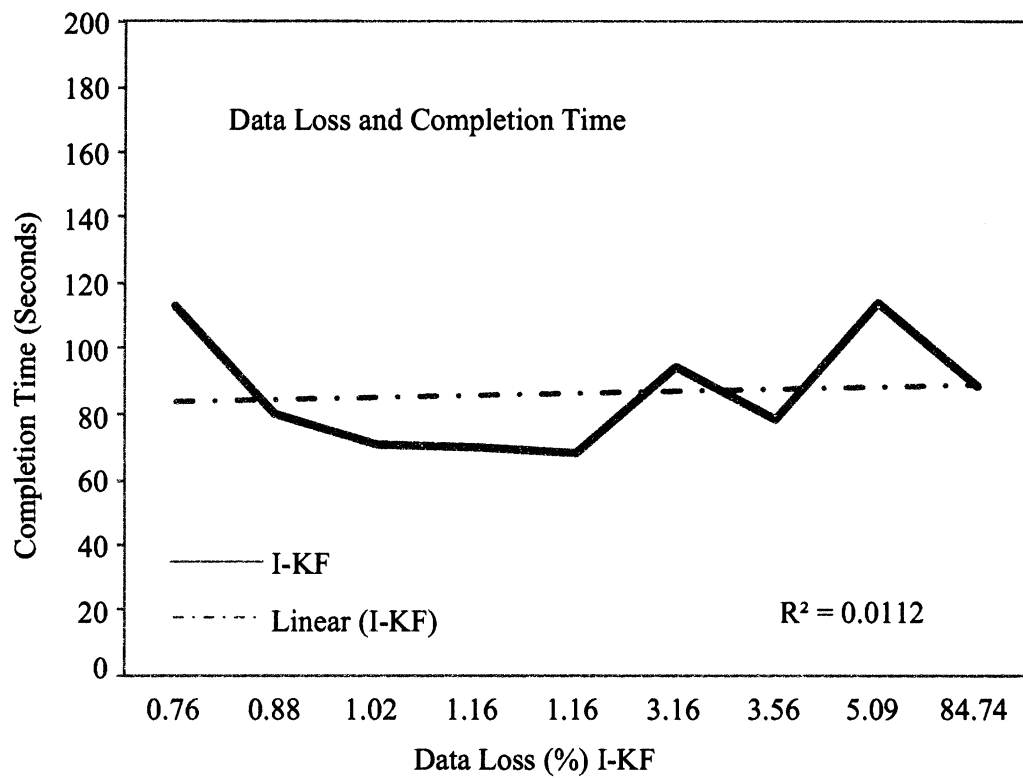


Figure 11. Completion Time vs. Data Loss for the I-KF. Represents recorded range for the I-KF model.

The performance I-KF results indicate very slow linear growth of average completion times with increased data loss. Remarkably, participants were able to perform well with the I-KF model even when data loss reached levels of more than 80%. With I-VT model, the data loss impact was much greater. Completion times were higher for the same level of data loss. It was not possible to interact with iGaze interface using the I-VT model when data loss exceeded 50% of the eye tracking data.

Questionnaire

Figure 12 presents the scores obtained for the device assessment questionnaire. According to the subjective evaluation, iGaze performance with the I-VT was perceived as smoother, induced less physical, eye and neck fatigue with more smoothness during operation than the iGaze with the I-KF. However, the iGaze with the I-VT caused greater shoulder fatigue.

On the other hand, the iGaze with I-KF required less mental effort for the operation. In addition, I-KF provided more accurate pointing, higher operation speed, and easier target selection. Furthermore, participants gave higher preference scores to the I-KF model when they compared the eye-gaze-driven interface to the interface controlled by keyboard/mouse. I-KF scored higher in general comfort evaluation. Participants also answered that the iGaze interface driven by the I-KF was easier to use when compared to the I-VT. The differences in the evaluation of the above mentioned categories were not statistically significant.

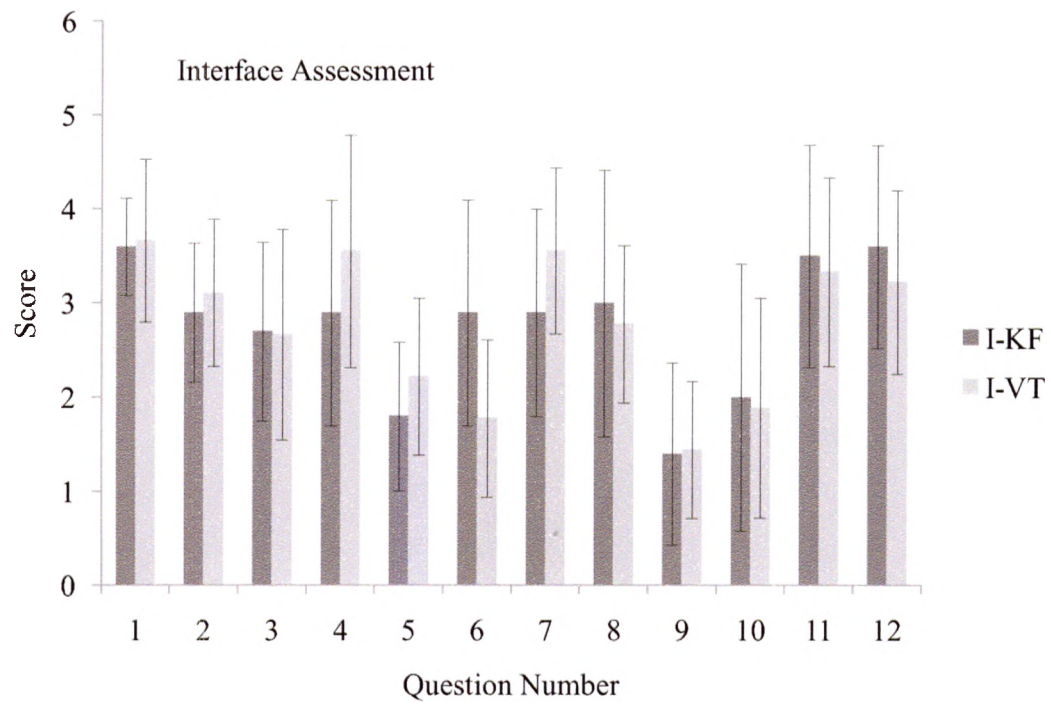


Figure 12. Device Assessment Questionnaire results.

CHAPTER VII

DISCUSSION

Based on our observations, participants who were wearing eye correcting devices did not perform well in cases when iGaze interface was controlled by the I-VT model. Very frequently, such participants had lower accuracy and higher data loss. However, I-KF performed much better in those cases due to the predictive and accuracy improvement capability of the filter.

In the current implementation of I-KF model, parameters such as threshold value for the chi-square test, sampling window size, and initial values for the system covariance matrix were empirically selected. In theory, such parameters depend on eye tracker sampling frequency and individual calibration accuracy. In the future, we would like to create a formula that would allow computing I-KF setup parameters based on experiment parameters.

The difference between subjective score values was not statistically significant with an exception on the question of general comfort and the difficulty of the interface use. The increase of the subject pool should address this issue.

CHAPTER VIII

CONCLUSION

This thesis has evaluated subjectively and objectively the performance of the Velocity Threshold (I-VT) and the Kalman Filter (I-KF) eye movement detection models. Both eye movement classification models were implemented with a standalone eye-gaze-driven photo viewing application (iGaze) and additionally tested with an accuracy test. The results of the evaluation indicate that on an average the I-KF allowed participants to complete more tasks with shorter completion time, while providing higher general comfort, accuracy, operation speed, and easier target selection than the I-VT model. The scores on the general comfort and the overall interface evaluation were also higher when the Kalman filter was employed.

The task completion time for the I-KF model did not decrease substantially in cases of decreased accuracy and high data loss. For the I-VT model, the task completion time increased when higher data loss and/or low accuracy of the equipment was reported. In two cases of extreme data loss (>50%), the iGaze application was unusable with the I-VT model, but when control was switched to the I-KF it was possible to complete assigned tasks.

Importantly, the accuracy verification test indicated that I-KF provided a 10% accuracy of increase when reporting locations of the detected fixations. This finding suggests that by employing the I-KF model in a design of an eye-gaze-guided system it

would be possible to improve the accuracy of the eye-tracker equipment itself, which is especially potent in cases of emerging low-cost eye trackers.

One of our motivations to create the iGaze application was to provide an access to a computer to people with disabilities. Based on the findings provided in this thesis, we can recommend the I-KF model to the designers of the eye-gaze-guided computer interfaces. The I-KF is a real-time detection algorithm based on matrix multiplication, which clearly improves system performance by increasing the accuracy of the eye-tracking device and providing eye-position data during eye tracker failures.

In our future work, we plan to create and evaluate more interactive applications and improve the robustness of the eye-gaze-guided computer applications by further improving the accuracy of the eye-gaze input.

APPENDIX A

DESIGN DESCRIPTION FOR iGAZE

Purpose

The following is a description of the software requirements of I-Gaze, a gaze-based digital photograph viewer.

Scope

iGaze is a stand-alone application which enables the user to view digital photographs using an eye-tracker and a mouse as inputs. The application maps eye-movements such as, but not limited to, dwells, fixations and blinks to actions that a user can carry out on a digital photo-album or a single photograph such as open, select and zoom. The application has a gaze-based user-interface and a middle-tier module to route information from the eye-tracker to the user-interface.

Overall description

Product perspective

iGaze provides functionality to view and manipulate digital photographs using eye-gaze. The application includes a complete set of APIs to enable future enhancements to the application.

Product functions

iGaze communicates with an eye-tracker via an event handler module to allow the user to select and open digital albums, and select and zoom, digital photographs.

User characteristics

The user has basic knowledge of how to use a personal computer. The user has basic training on using eye-tracker technology.

Assumptions and dependencies

The application is run on Windows XP operating system. Eye-gaze data is gathered by Tobii X120 eye tracker. The user will use a 19'' monitor. The application will handle image files with the JPG, .JPEG, GIF, PNG and BMP extensions.

GUI

Microsoft Expression Blend is one of the main tools used for developing the GUI.

Programming tools

Visual Studio 2005 and Microsoft Expression Blend are the two programming tools used.

Key Definitions

Gaze - A signal from 1 or more eyes is detected without any intentional interruption.

Dwell - A fixation is detected for a certain period of time.

Blink - The signal from both eyes is intentionally interrupted for a certain duration, and then resumed, during a fixation.

Left-Wink - The signal from the left eye is intentionally interrupted but the signal from the right eye is still present.

Right-Wink - The signal from the right eye is intentionally interrupted but the signal from the left eye is still present.

Look-Away Up/Down/Left/Right - The signal from both eyes is intentionally interrupted for a certain duration at a point close to an edge of the screen, and then resumed, during a saccade.

Picture - An image that has a similar appearance to some subject- usually a physical object or a person

Album - An album is a collection of pictures.

Filmstrip - A Filmstrip is a length of film containing a series of pictures prepared for still projection.

Thumbnails - A reduced image of pictures, used in order to view multiple images on a screen simultaneously.

Use Cases

Gaze Interaction

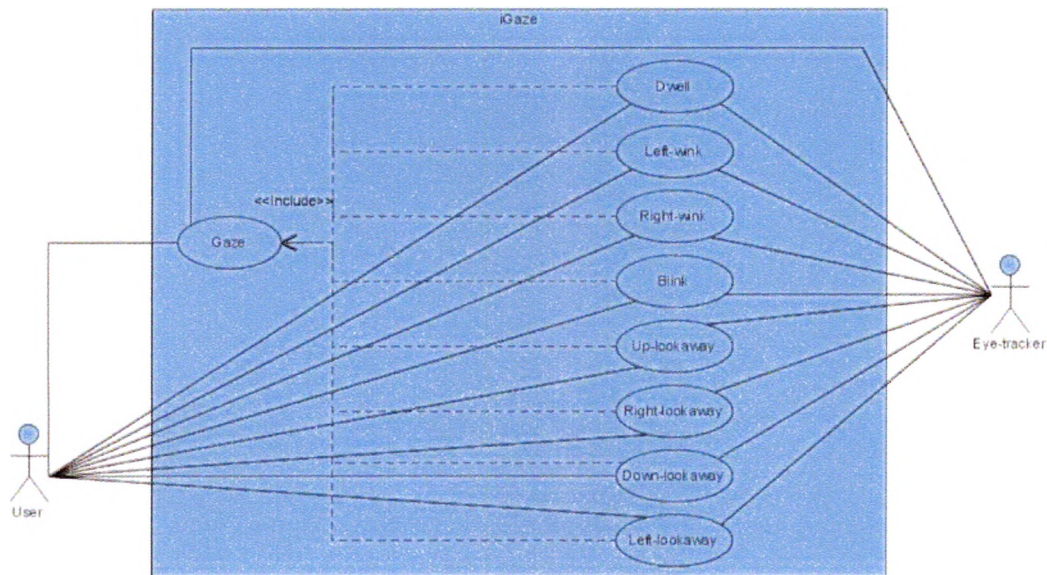


Figure 13. Gaze Interaction Diagram. Functionalities that are possible by dwell time interaction.

Brief description:

The user makes a gaze action to interact with the application.

Initial step-by-step description:

For this use case to be active, the application has to receive the gaze data from the eye tracker. Based on the gaze that the user makes, the gaze will be classified into one of these type – Dwell, Left-wink, Right-wink, Blink, Up-look away, Down-look away, Right-look away, Left-look away.

Select Album

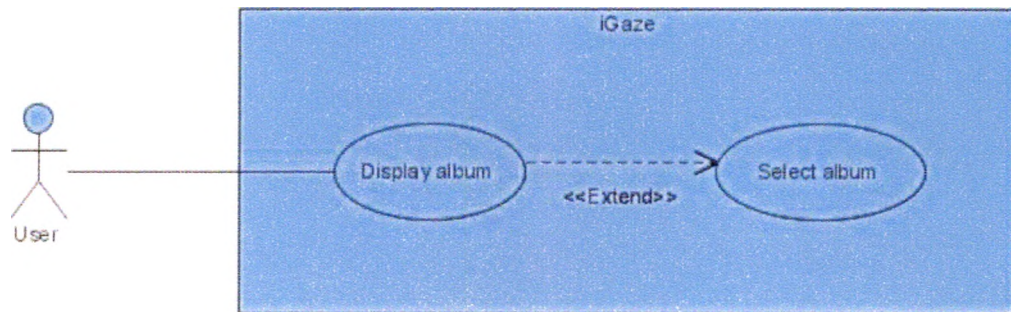


Figure 14. Select Album. The use case for select album functionality.

Brief description:

The user selects an album in the application.

Initial step-by-step description:

For this use case to be active, the user's gaze has to be on one of the albums in the application. The use case extends from the Display album use case.

Select and Display Album

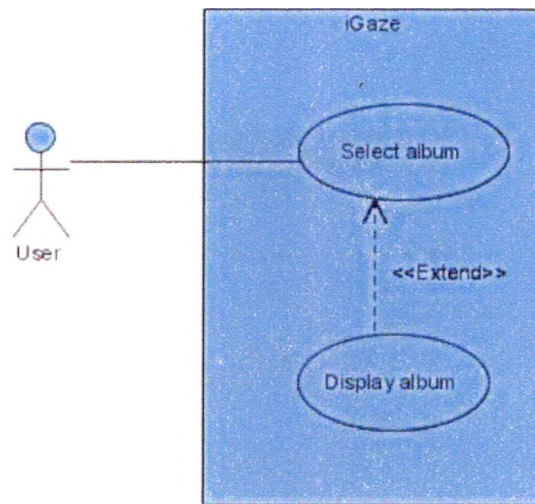


Figure 15. Select and Display Album. The use case for selecting and displaying an album.

Brief description:

The user selects an album in the application, which in turn gets displayed.

Initial step-by-step description:

For this use case to be active, the user's gaze has to be on one of the albums in the application. The use case extends from the Display album use case, which implies that an album is displayed when a User selects an album.

Select and Display Picture

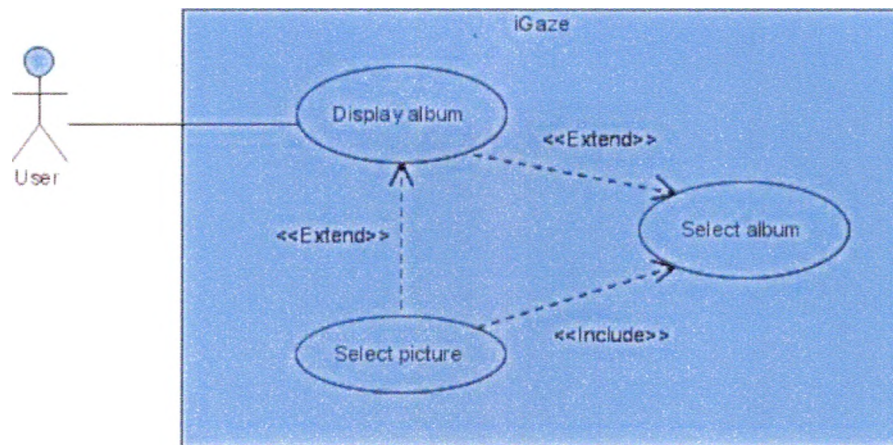


Figure 16. Select and Display Picture. The use case for selecting and displaying a picture.

Brief description:

The user selects a picture from the displayed album in the application.

Initial step-by-step description:

For this use case to be active, the user's gaze has to be on one of the pictures in displayed album. The use case extends from the Display album use case, and also includes the use case Select album function. The Select picture can happen only after the Select and Display album function occurs.

Select Picture

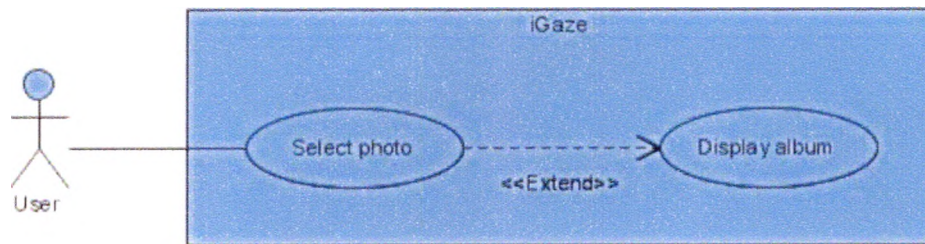


Figure 17. Select Picture. The use case for selecting a picture.

Brief description:

The user selects a picture from the list of pictures displayed in the displayed album.

Initial step-by-step description:

For this use case to be active, the user's gaze has to be on one of the pictures in the displayed album. The use case extends from the Display album use case, which implies that a picture can be selected when a User selects and displays an album.

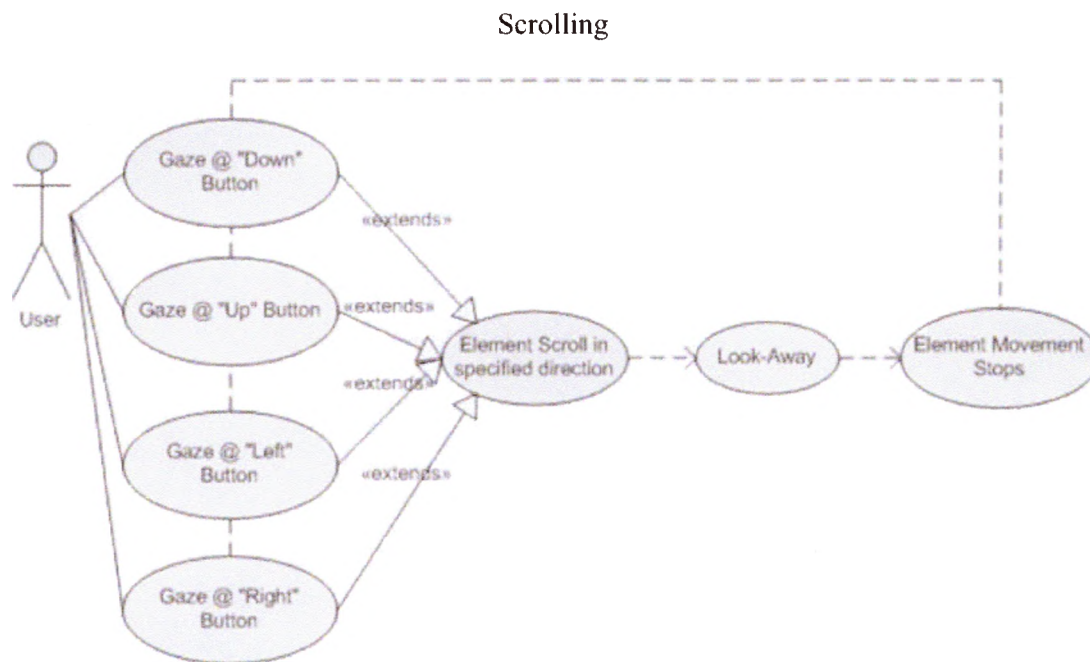


Figure 18. Scrolling. The use case for scrolling functionality.

Brief description:

The user gazes on the scroll items in the application, to either scroll up or down, or scroll left or right.

Initial step-by-step description:

This use case will be active only when the User wishes to perform a scrolling functionality. There are four possible scroll directions all of which extend from the Element Scroll in specified direction. The scrolling stops when the User's gaze is no longer on the scrolling element or when the eye movement detection stops.

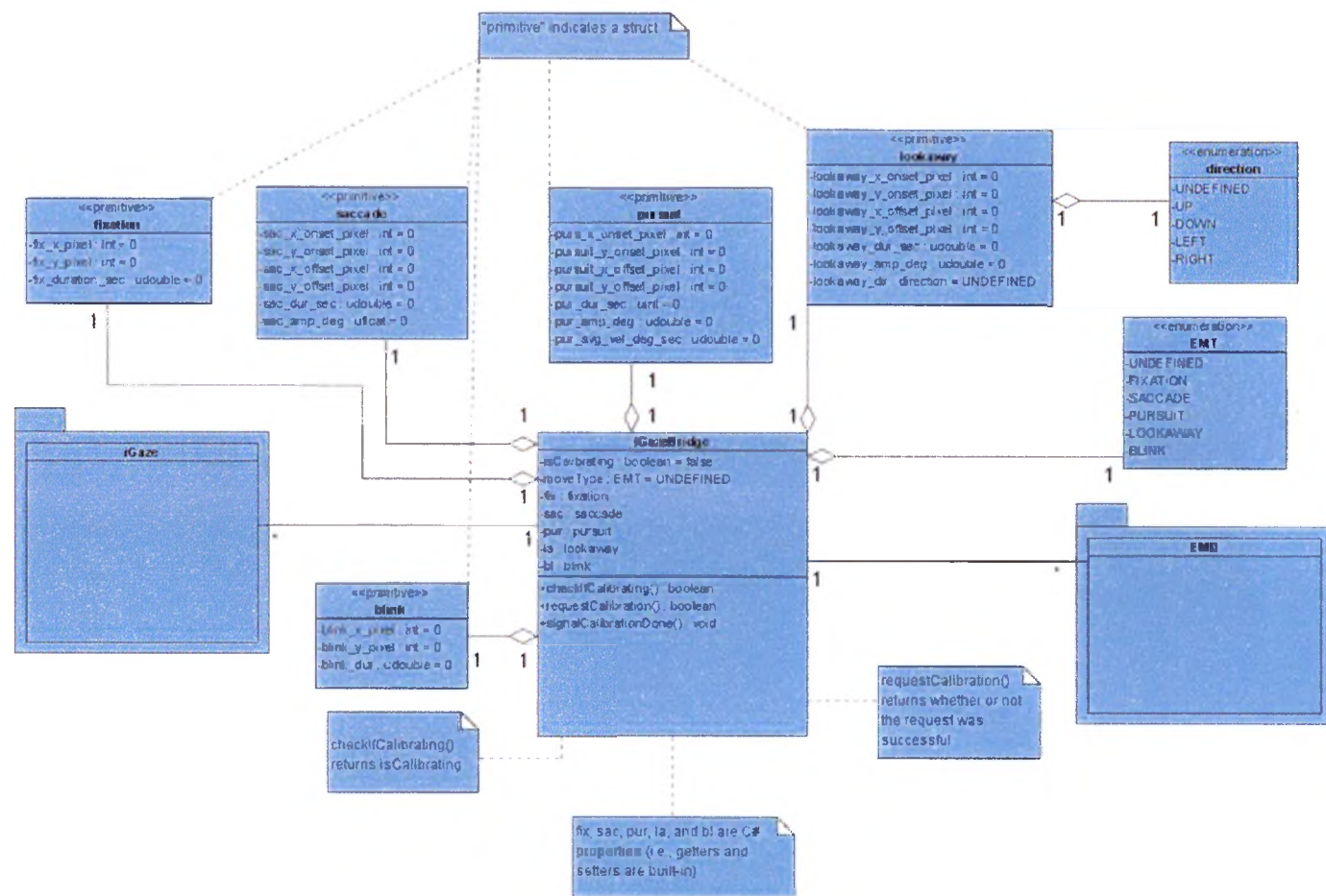


Figure 19. iGaze Overview. The class diagram of the iGaze application.

APPENDIX B

TECHNICAL AND IMPLEMENTATION DETAILS OF iGAZE

Technical Details

The iGaze application was developed using Microsoft .NET 3.0 Components. The GUI interface used the Windows Presentation Foundation Class components. The event handling code for GUI is written in C#.

Figure 20 shows the framework of the iGaze application. The “Device specific code” is provided by the vendor Tobii. This provides us an access mechanism to the data from the eye tracker “Tobii X120”. The data from the eye tracker are used by the “Eye movement classification code” which has holds both of the Eye Movement Classification models the I-VT and the I-KF. The classification models make use of the “Eye Movement Library”, which has an extensive list of variables and parameters (See Figure 19). Once the classification is done, the data is pushed on to the “Bridge / Data pool”. The data from the bridge are then read by the “iGaze interface” to provide feedback and perform actions like selection.

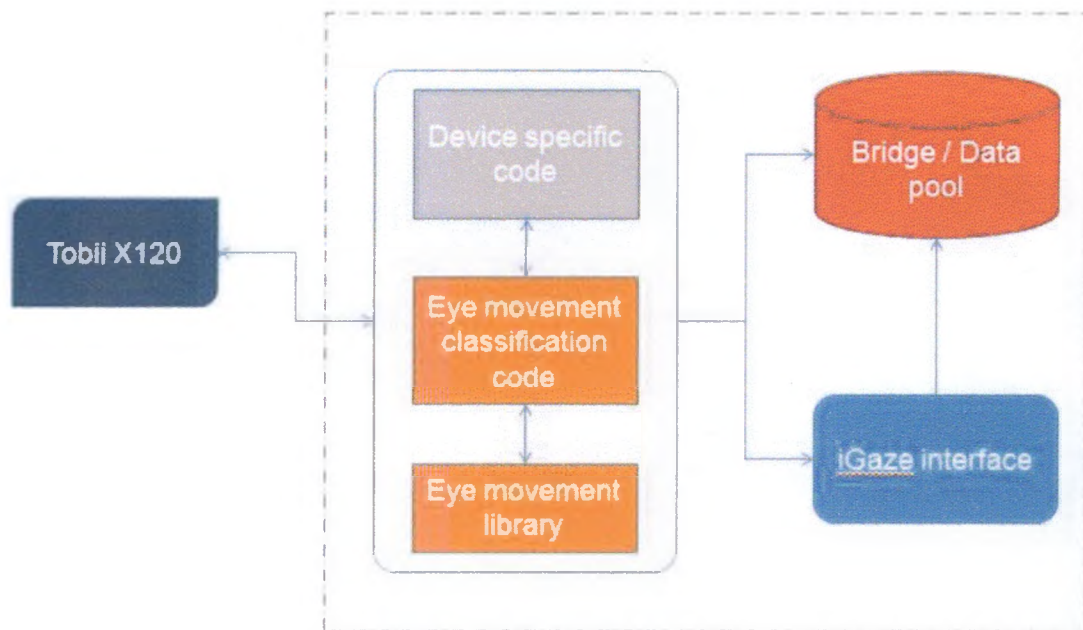


Figure 20. iGaze Framework. Overview of the iGaze framework.

Implementation Details

Window1: MAIN CLASS

This class has all the data members and 2 lists albums and images declared.

WINDOW1 CONSTRUCTOR

The constructor initializes the following:

- path is set to My pictures.
- albums and images are initialized to empty lists.
- number of albums that is displayed at once in the left= 5.
- images of a selected album displayed as thumbnails at a time= 12.
- images displayed as filmstrip below at a time = 5.
- sets the last loaded album index and the last loaded image index.

MYIMAGE : SUB CLASS

this class has the data members _image, _name and _index.

the get and set functions return the current image, its name and the index.

ALLIMAGES

This is a function for windows1 class.

It returns a list of pictures in the selected album path.

Direction denotes which index from the file listing would we load from in the resulting array.

The images can only be JPG.

IMAGES GET FUNCTION used for binding images in the list

MYALBUMS CLASS

This class has data members _image and _name.

loads a dummy bitmap image for the album icon.

the get and set functions return the image and the name of the album.

ALLALBUMS

This function returns a list of albums in a given path.

Direction is used to determine which index the albums will get from the listing array to the result array.

ALBUMS : GET FUNCTION returns album

ALBUMLIST_SELECTIONCHANGED -- EVENT HANDLER

Action: Album is selected from the albumlist-

Result: All the images of the current album are loaded as thumbnails in the thumbnail area.

DOWN_CLICK -- EVENT HANDLER

Action: Down button is selected in the album listing area-

Result: The next 5 albums that are below are displayed.

UP_CLICK -- EVENT HANDLER

Action: Up button is selected in the album listing area-

Result: The next 5 albums that are above are displayed.

IMAGEUP_CLICK -- EVENT HANDLER

Action: Up button is selected in the thumbnails listing area-

Result: The next 16 images that are above are displayed.

IMAGEDOWN_CLICK -- EVENT HANDLER

Action: Down button is selected in the thumbnails listing area-

Result: The next 16 images that are below are displayed.

IMAGELIST_SELECTIONCHANGED -- EVENT HANDLER

Action: When a new image is selected from the thumbnails-

Result: The selected image appears at the centre.

All the images of the current album is loaded as a strip at the bottom.

The Album grid is hidden.

ALBUMVIEW_CLICK -- EVENT HANDLER

Action: When a new album is selected, click happens here.

Result: All the 16 images of the current selected album is displayed in the thumbnail area.

SCROLLLEFT_CLICK -- EVENT HANDLER

Action: Scrolled to left-

Result: The images to the left in the Strip are loaded.

SCROLLRIGHT_CLICK -- EVENT HANDLER

Action: Scrolled to left-

Result: The images to the left in the Strip are loaded.

PICTURELIST_SELECTIONCHANGED -- EVENT HANDLER

Action: Image from the strip is selected

Result: The Image appears in the centre of the screen.

URITOTHUMBNAILCONVERTER

Converts from bitmap to thumbnail when the picture is selected from album.

BIBLIOGRAPHY

- Brown, R. G., & Hwang, P. Y. C. (1996). *Introduction to random signals and applied kalman filtering with matlab exercises and solutions* (3rd ed.). New York: Wiley.
- Douglas, A., Kirkpatrick, A. E., & MacKenzie, I. S. (1999). Testing pointing device performance and user assessment with the ISO 9241, part 9 standard. Paper presented at the *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, Pennsylvania, United States. 215-222.
- Duchowski, A. T. (2007). *Eye tracking methodology: Theory and practice* (2nd ed.) Springer.
- International Standards Organization (2000). *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 9: Requirements for non-keyboard input devices*. ISO 9241-9:2000. Geneva, Switzerland: International Standards Organization.
- Irwin, D. E. (1992). Visual memory within and across fixations. Paper presented at the *In Eye Movements and Visual Cognition: Scene Perception and Reading*, 146-165.
- Komogortsev, O. V., & Khan, J. (2007). Kalman filtering in the design of eye-gaze-guided computer interfaces. *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments* 679-689.
- Komogortsev, O. V., & Khan, J. I. (2008). Eye movement prediction by kalman filter with integrated linear horizontal oculomotor plant mechanical model. Paper presented at the *ETRA '08: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, Savannah, Georgia. 229-236.
- Kumar, M., Klingner, J., Puranik, R., Winograd, T., & Paepcke, A. (2008). Improving the accuracy of gaze input for interaction. Paper presented at the *ETRA '08: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, Savannah, Georgia. 65-68.
- Kumar, M., Paepcke, A., & Winograd, T. (2007). EyePoint: Practical pointing and selection using gaze and keyboard. Paper presented at the *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, San Jose, California, USA. 421-430.

- Leigh, R. J., & Zee, D. S. (2006). *The neurology of eye movements* (3rd ed.) Oxford University Press, UK.
- Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. Paper presented at the *ETRA '00: Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, Palm Beach Gardens, Florida, United States. 71-78.
- Sauter, D., Martin, B. J., Di Renzo, N., & Vomscheid, C. (1991). Analysis of eye tracking movements using innovations generated by a kalman filter. *Medical & Biological Engineering & Computing*, 29(1), 63-69.
- Sibert, L. E., & Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. Paper presented at the *CHI '00: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, The Hague, The Netherlands. 281-288.
- Tien, G., & Atkins, M. S. (2008). Improving hands-free menu selection using eyegaze glances and fixations. Paper presented at the *ETRA '08: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, Savannah, Georgia. 47-50.
- Urbina, M. H., & Huckauf, A. (2007). Dwell time free eye typing approaches. *COGAIN*, Leicester, UK. 65-70.
- Yarbus, A. L. (1967). *Eye Movements and vision* (B. Haigh Trans.). New York: Plenum Press.
- Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. Paper presented at the *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, Pennsylvania, United States. 246-253.
- Zhang, X., & MacKenzie, I. S. (2007). Evaluating eye tracking with ISO 9241 - part 9. *Proceedings of HCI International 2007*, 779-788.

VITA

Sandeep Munikrishne Gowda was born in Bangalore City, Karnataka, India on May 16, 1984, the son of Munikrishne Gowda and Ananda Munikrishne Gowda. He grew up in Bangalore, India, where he obtained his pre-university college degree in 2002. From 2002 to 2006, he attended East Point College of Engineering & Technology (Affiliated to Visvesvaraya Technological University, Belgaum), Bangalore, where he obtained the degree of Bachelor of Engineering in Information Science. He subsequently worked as a web designer for a year in Bangalore, India. He entered The Graduate College of Texas State University-San Marcos as a full-time student in the Fall of 2007.

