Integrating Eye Gaze into Pen-Based Systems

by

Çağla Çığ Karaman

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Doctor of Philosophy

in

Computer Engineering

August 14, 2017

Integrating Eye Gaze into Pen-Based Systems

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Çağla Çığ Karaman

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Tevfik Metin Sezgin

_____

Assoc. Prof. Yücel Yemez

_____

Prof. Zehra Çataltepe

_____

Assoc. Prof. Albert Ali Salah

_____

Assist. Prof. Ayşe Küçükyılmaz

Date: _____

This thesis is dedicated to my son, Ateş.

"Yours is the light by which my spirit's born: – you are my sun, my moon, and all my stars." – E.E. Cummings

ABSTRACT

In typical human-computer interaction, users convey their intentions through traditional input devices (e.g. keyboards, mice, joysticks) coupled with standard graphical user interface elements. Recently, pen-based interaction has emerged as a more intuitive alternative to these traditional means. However, existing pen-based systems are limited by the fact that they rely heavily on auxiliary mode switching mechanisms during interaction (e.g. hard or soft modifier keys, buttons, menus). In this thesis, we describe how eye gaze movements that naturally occur during pen-based interaction can be used to reduce dependency on explicit mode selection mechanisms in pen-based systems. In particular, we show that a range of virtual manipulation commands, that would otherwise require auxiliary mode switching elements, can be issued with an 88% success rate with the aid of users' natural eye gaze behavior during pen-only interaction. We believe the non-intrusive and transparent use of gaze modality as a complementary information channel will bring us closer to the goal of truly intuitive pen-based interaction.

To this end, we (1) investigate the nature of gaze behavior during various pen-based interaction scenarios, (2) mine for, extract, and create statistical models for useful and usable gaze behavior patterns while users keep their normal habits and ways to interact, (3) use these models to create fully integrated gaze-based intelligent information visualization systems that are able to dynamically adapt to user's spontaneous task-related intentions and goals, and (4) evaluate these systems via a thorough usability study involving 19 participants and 5 different user interface scenarios. Evaluation results demonstrate that we can successfully establish a shared understanding between the user and the adaptive interface based on users' natural eye gaze behavior, and without interrupting the interaction flow.

ÖZETÇE

Alışılagelmiş insan-bilgisayar etkileşiminde, kullanıcılar niyetlerini geleneksel giriş cihazları (ör. klavye, fare, oyun çubuğu) ve bunlarla uyumlu standart grafiksel kullanıcı arayüzü elemanları ile iletmektedir. Yakın zamanda kalem-temelli etkileşim, bu geleneksel yöntemlere daha doğal bir alternatif olarak öne çıkmıştır. Ancak, mevcut kalem-temelli sistemler de etkileşim süresince yardımcı mod değiştirme mekanizmalarına (ör. fiziksel/sanal tuşlar, düğmeler, menüler) olan yüksek bağımlılıkları nedeniyle kısıtlı kalmaktadır. Bu tezde, kalem-temelli etkileşime doğal olarak eşlik eden göz bakış hareketlerinin, kalem-temelli sistemlerin yardımcı mod değiştirme mekanizmalarına olan bağımlılıklarını azaltmak için nasıl kullanılabileceği tarif edilmektedir. Özellikle, normalde yardımcı mod değiştirme elemanlarına gereksinim duyan bir takım sanal manipülasyon komutunun, kullanıcıların doğal göz bakış davranışları yardımıyla %88 başarı oranıyla iletilebildiği gösterilmektedir. Bakış kipinin müdahalesiz ve şeffaf bir şekilde tamamlayıcı bilgi kanalı olarak kullanımının bizi gerçek manada doğal kalem-temelli etkileşim hedefine yaklaştıracağına inanmaktayız.

Çalışmamızda, (1) çeşitli kalem-temelli etkileşim senaryolarında bakış yönü davranışının doğası incelenmiş, (2) kullanıcılar normal alışkanlıklarını ve etkileşim yollarını devam ettirirken kullanışlı ve kullanılabilir bakış yönü davranış örüntüleri çıkarılmış ve istatistiksel yöntemlerle modellenmiş, (3) bu modeller, kullanıcının komutlarla alakalı anlık niyet ve hedeflerine dinamik olarak adapte olabilen tamamıyla entegre bakış yönü-temelli akıllı bilgi görselleştirme sistemleri yaratmak için kullanılmış, (4) bu sistemler, 19 katılımcı ve 5 farklı kullanıcı arayüzü senaryosu içeren kapsamlı bir kullanılabilirlik çalışması ile değerlendirilmiştir. Değerlendirme sonuçları, kullanıcıların doğal göz bakış davranışlarını kullanarak ve etkileşim akışını bozmadan kullanıcı ve adaptif arayüz arasında ortak bir anlayışı başarıyla sağlayabildiğimizi göstermiştir.

ACKNOWLEDGMENTS

Foremost, I would like to express my deepest gratitude to my advisor, Prof. Tevfik Metin Sezgin, for accepting me to Intelligent User Interfaces Lab, supporting me through thick and thin, constantly challenging me to be a better researcher, and always guiding me in the right direction. I am forever indebted to you.

I would like to thank Prof. Yücel Yemez and Prof. Zehra Çataltepe for agreeing to be a part of my thesis committee. I would like to thank them not only for their time and patience, but also for their intellectual contributions to my development as a researcher.

I would like to thank my husband, Bünyamin Karaman, for seeing me for who I am and always finding a way to be at my side. I cannot imagine a world without you. I would like to thank my mom, dad, and brother for teaching me that love is all that matters, and inspiring me to be a better person. I would like to thank my extended family, especially my mother-in-law, for always looking out for me and my little family. I would also like to honor the loving memory of my grandmother and aunt. You both are deeply missed.

Finally, I would like to thank the members of the Intelligent User Interfaces Lab: Banuçiçek Gürcüoğlu, Burak Özen, Şerike Çakmak, Cansu Şen, Neşe Alyüz, Özem Kalay, Atakan Arasan, Kemal Tuğrul Yeşilbek, Ozan Can Altıok, Bekir Berker Türker, Erelcan Yanık, Ayşe Küçükyılmaz, Senem Ezgi Emgin, Sinan Tümen, Çağlar Tırkaz. You have supported me in so many different ways and I'm glad we always found a way to laugh together.

This thesis is funded by TÜBİTAK (The Scientific and Technological Research Council of Turkey) under grant numbers 110E175 and 113E325 and TÜBA (Turkish Academy of Sciences).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

2D	Two-Dimensional
3D	Three-Dimensional
AOI	Area of Interest
ANOVA	Analysis of Variance
API	Application Programming Interface
AUC	Area Under Curve
BIC	Bayes Information Criterion
CPU	Central Processing Unit
DBN	Dynamic Bayesian Network
DTW	Dynamic Time Warping
EER	Equal Error Rate
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HMM	Hidden Markov Model
HSD	Honest Significant Difference
IDE	Integrated Development Environment
IDM	Image Deformation Model
LED	Light-Emitting Diode
M	Mean
mRMR	Minimum Redundancy Maximum Relevance
NASA-TLX	NASA Task Load Index
ROC	Receiver Operating Characteristic
PC	Personal Computer

RAM	Random Access Memory
RBF	Radial Basis Function
SD	Standard Deviation
SDK	Software Development Kit
SUS	System Usability Scale
SVM	Support Vector Machine
UI	User Interface
UP-GMA	Unweighted Pair Group Method with Arithmetic Mean
URL	Uniform Resource Locator
WIMP	Windows, Icons, Menus, Pointer

Chapter 1

INTRODUCTION

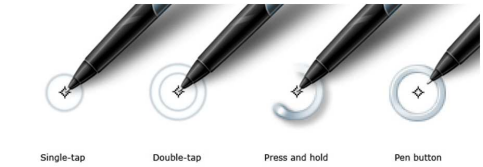
In pursuit of an invisible interface for the computer of the 21st century [81], the explicit human-computer interaction model is gradually being replaced by the implicit interaction model. This can be observed in the shift from text terminals and graphical user interfaces to ambient interfaces and proactive personal assistants. Implicit interfaces sense and reason about user actions (that are not primarily aimed to interact with a computerized system) to automatically trigger appropriate reactions [69]. In order to reason about user actions with innovative sensors like eye trackers, implicit interfaces model human behavior by extracting useful and usable patterns while users keep their normal habits and ways to interact. The advantage of implicit interfaces is that the users do not need explicit commands, prior knowledge, or training to interact with the system.

Shortcomings of the command-based explicit interaction model are especially highlighted in mobile computing systems where the ability to input commands is limited by the same compact form factors that make new generation mobile devices portable (e.g. screen size limitations, absence of a physical mouse/keyboard). Designed well, intelligent user interfaces for mobile devices can assist the users by implicitly generating commands based on previously learned models of human behavior patterns.

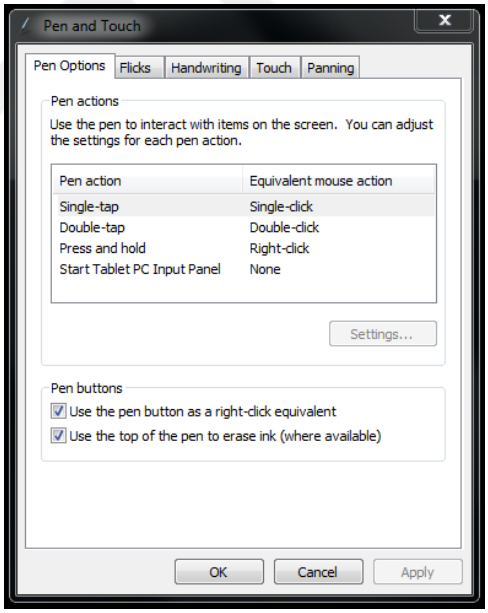
Well-designed intelligent user interfaces can especially profit pen-based mobile devices. These devices have emerged to offer a more intuitive interaction alternative through the pen-and-paper metaphor, but failed to do so by insisting on user interfaces that emulate traditional explicit interaction. In other words, despite what their name suggests, pen-based devices are not purely pen-based [61].

(a) Switching between pen and multi-touch input for object manipulation (e.g. image resizing) in pen-based smart phones.

(b) On-pen or on-tablet external buttons in pen-based graphics tablets.



(c) Various tapping and/or holding techniques to access context/pop-up menus in pen-based tablet computers.



(d) The pen is used to emulate a mouse in pen-based tablet computers.

Figure 1.1: Some examples to pen-based devices with interaction paradigms that are not purely pen-based. Gaze-based prediction of virtual interaction tasks in pen-based interaction is a step towards systems that require fewer mode changes [47].

For example, in pen-enabled smart phones, many actions force the user to put the pen aside and switch to multi-finger gestures (e.g. spread/pinch for zoom in/out, and swipe to navigate back/forward). These gestures require the simultaneous use of 2, 3, or even 4 fingers (Fig. 1.1a). The necessity of switching between pen and multi-touch input goes against the goal of seamless interaction in pen-based devices.

Even the state-of-the-art devices and software specifically built for pen-based interaction lack purely pen-based interaction. For example, graphics tablets preferred mainly by digital artists such as Wacom Cintiq 24HD (Fig. 1.1b) are often referred to as "heaven on earth" by users. However, even with these high-end models many tasks are still accomplished via on-pen or on-tablet external buttons called "express keys", "touch rings", and "radial menus". These buttons allow the user to simulate keystrokes including letters, numbers, and modifier keys (e.g. Shift, Alt, and Control). To issue a virtual manipulation command (e.g. scroll), the user has to locate the correct button which interrupts the interaction flow, hence causing an overall disappointing experience.

Another example where we lose purely pen-based interaction is with tablet computers. In most pen-based applications, features are hidden in standard context/pop-up menus that are accessed via tapping and/or holding the pen on the tablet screen in various ways (Fig. 1.1c). In this case, the pen is used to trigger mouse clicks, which fits the traditional GUI/WIMP-based interaction paradigm, rather than that of a purely pen-based interaction (Fig. 1.1d).

These issues show that existing pen-based systems depend substantially on multi-finger gestures, context/pop-up menus, and external buttons which goes against the philosophy of pen-based interfaces as a more intuitive interaction alternative. In this thesis, we show that eye gaze movements that naturally accompany pen-based user interaction can be used to infer a user's task-related intentions and goals. The non-intrusive and transparent use of eye gaze information for task prediction brings us closer to the goal of purely pen-based interaction and reduces the reliance on multi-finger gestures, context/pop-up menus, and external buttons.

Our approach consists of tracking eye gaze movements of the user during pen-based interaction and fusing the spatio-temporal information collected via gaze and sketch modalities in order to predict the current intention of the user. We use the term "intention" to refer specifically to the intention of the user to issue a virtual manipulation command. Virtual manipulation commands that we can currently predict are: *drag*, *maximize*, *minimize*, and *scroll*. Additionally, we can distinguish whether the user intends to issue any of these virtual manipulation commands, or intends to sketch using our special-purpose task class called *free-form drawing*.

In addition, we propose an intelligent system capable of actively monitoring user's eye gaze and pen input to detect the intention to switch modes in an online setting, and act accordingly. When the user performs a pen action (demarcated by a pen-down and a pen-up event), our intelligent user interface actively detects and switches to the currently intended mode of interaction based on user's synchronized pen trajectory and eye gaze information during pen-based interaction. Intention predictions are carried out by the previously trained model and the features extracted from the corresponding sketch-gaze data of the user.

Furthermore, we present a gaze-based biometric authentication system that uses the same set of tasks we employ for intention prediction. Via these tasks, users introduce themselves and gain authorized access to their mobile devices. Our authentication system depends on strongly unforgeable behavioral characteristics of an individual as opposed to more easily forgeable characteristics based on possession, knowledge, or biometrics. Accordingly, it can be used on its own or as an extra layer of identity verification in both high-security (e.g. forensics, access restriction, e-commerce) and low-security (e.g. customized user profiles, adaptive user interfaces) scenarios.

Chapter 2 gives an outline of the state-of-the-art gaze-based interaction and biometric authentication studies in a categorical manner with relevant examples for each category. Chapters 3 and 4 respectively describe how we build our gaze-based virtual task prediction system and how we integrate our prediction system to build a

gaze-based intelligent user interface. Chapter 5 elaborates on our gaze-biometric authentication system. Chapter 6 concludes with a discussion of our work, and Chapter 7 gives a summary of future directions.

Chapter 2

LITERATURE REVIEW

2.1 Gaze-Based Interaction

State-of-the-art gaze-based interaction studies fall under two main categories depending on the style of interaction: *explicit gaze-based interaction studies* and *implicit gaze-based interaction studies*.

Explicit gaze-based interaction studies are based on the eye-mind hypothesis, in which, intentional eye movements are associated with interface actions [6]. In other words, in explicit gaze-based interfaces, gaze is employed as an explicit pointing device. This requires the gaze to be used for manipulation in addition to its natural purpose, visual perception. This approach forces the user to be aware of the role of the eye gaze and therefore results in high cognitive workload [3]. Zhai et al. argue that "other than for disabled users, who have no alternative, using eye gaze for practical pointing does not appear to be very promising" [86]. Kumar et al. agree that "overloading the visual channel for a motor control task is undesirable" [45]. In line with these arguments, our work avoids forcing the user to consciously adopt unnatural gaze behavior for interaction purposes and instead uses gaze movements that naturally accompany manipulation tasks for building a gaze-based intelligent user interface.

In implicit gaze-based interaction studies, the computer system passively and continuously observes the user in real-time and provides appropriate responses. In order to provide satisfying and natural responses, the computer system must be able to infer user's intentions from his/her spontaneous natural behaviors. An intention can be, for instance, moving a window, scrolling a piece of text, or maximizing an image [6]. Studies [3, 9, 11, 32, 35, 84] provide qualitative observations and quantitative evidence

suggesting that well-structured tasks have unique eye movement signatures. However, the majority of the related work on implicit gaze-based interaction focuses solely on post-hoc analysis of eye movements collected during natural interaction. There are only a few researchers who have made considerable attempts at interpreting user behavior for online task prediction - and even fewer researchers who have employed their online task prediction system to build an intelligent user interface. Therefore, implicit gaze-based interaction studies can be grouped under three subcategories[1]: ***gaze-based task analyzers***, ***gaze-based task predictors***, and ***gaze-based intelligent user interfaces***. First two categories can be further divided as ***daily task analyzer/predictors*** and ***virtual task analyzer/predictors*** depending on the nature of tasks taken into consideration. Daily tasks such as sandwich making and stapling a letter are ordinary activities in everyday settings [27, 46, 83–85] whereas virtual tasks such as reading an electronic document and manipulating a virtual object involve the use of a computer system [2, 3, 6, 11, 30]. We focus on pen-based tablet devices; therefore, we are interested in analyzing, predicting, and building an intelligent user interface for the range of virtual interaction tasks commonly performed on tablet devices. However, for completeness sake, our literature review covers daily tasks as well. In the following sections we provide a review of the related work that fall under each category. A more comprehensive summary of related work on task analyzers/predictors can be found in Appendix A.

2.1.1 Gaze-Based Task Analyzers

Daily Task Analyzers

Daily task analyzers focus on analyzing various characteristics of eye movements while users perform daily tasks. Land and Hayhoe [46] investigate the relationships between eye and hand movements in food preparation tasks such as brewing tea and fixing

[1]These subcategories are in line with Schmidt's categorization of implicit interaction into three basic building blocks as (1) perception of the user via sensors, (2) mechanisms to understand the sensor data, and (3) context-enabled applications [69].

a sandwich. Their results are largely composed of plots displaying how body, eye, and hand movements change in time and the authors point out that the control of eye movements is primarily directed at the ongoing motor actions. Yi and Ballard [83] also focus on the sandwich making task; however, they take a more probabilistic approach. The authors manually segment the task into subtasks such as locating the bread, spreading jelly on the bread, etc., and then use a dynamic Bayesian network (DBN) to model the task. However, the authors do not assess the goodness of their predictions and only provide graphs visualizing the real and inferred timings of the subtasks. Lastly, Hayhoe and Ballard [32] present a review of approaches that analyze eye movements in everyday visually guided behaviors, however these do not interpret user behavior for online task prediction.

Virtual Task Analyzers

Virtual task analyzers focus on analyzing various characteristics of eye movements for tasks that involve the use of a computer system. Iqbal and Bailey [35] study eye gaze patterns in four different tasks: reading comprehension, mathematical reasoning, search, and object manipulation. The authors segment the virtual interaction area into interface-specific areas of interest (AOI) and qualitatively inspect the relationship between amount of time spent on each AOI and task type. They show that the percentage of time spent on each AOI varies across task categories. However, they do not provide statistical analysis or a mechanism for prediction. Alamargot et al. [2] provide a detailed description of the eye movement characteristics recorded during reading and writing on a digital tablet. Gesierich et al. [30] observe proactive (i.e. anticipatory) eye movements in both action execution and action observation during a two-user virtual block stacking task. Our work differs from the listed virtual task analyzers in its successful attempt at interpreting user behavior for online task prediction.

2.1.2 Gaze-Based Task Predictors

Daily Task Predictors

Yu and Ballard [84, 85] use Hidden Markov Models (HMM) to discriminate between the tasks of unscrewing a jar, stapling a letter, and pouring water. In addition to features related to the eye, head, and hand movements, the authors also employ features related to scene objects being fixated by the user during task execution. This substantially simplifies task prediction as it practically reduces to discriminating between a jar, a stapler, and a carafe. Similarly, in more recent work, Fathi et al. [27] use Support Vector Machines (SVM) to discriminate numerous subtasks of a meal preparation task. The authors extract gaze-related features and features from the fixated scene objects that mainly describe their key visual properties. More specifically, Yu and Ballard [84, 85] demonstrate that the user most probably has an intention to staple a letter and not unscrew a jar or pour water if the object of focus is a stapler. However, it is certainly less clear whether the user intends to drag, maximize, minimize, scroll, or sketch on a virtual object; for instance, an image, even if the object of focus is the image itself. The ultimate aim of task predictors is not predicting the current task in a certain context but doing so in a context-independent way. All pieces of work focus on daily task prediction and utilize context information. This is feasible in the context of daily tasks; however, not as much so in the case of virtual tasks since the user might be performing a variety of different tasks while focusing on the same virtual object.

Bulling et al. [9, 10] present two closely related works. The authors focus on classifying tasks in three domains: reading, office activities, and cognitive psychology (more specifically, visual memory recall). The domain most related to our work is office activities consisting of copying, reading, writing, watching a video, and browsing the Web. They report an average precision score of 76.1% using SVMs, which is comparably higher than scores reported by related works that focus on gaze-based task prediction. A more recent closely related work is by Ogaki et al. [55]. They study the same indoor office activities as Bulling et al. [9, 10] and demonstrate that coupling

eye movements with ego-motion leads to better task classification performance. Their work resembles our work in its use of multiple modalities, however the features used in these studies depend heavily on preset templates to track repetitive patterns of eye movements and on constants for defining threshold levels, sliding window sizes, etc. On the contrary, the highly generic features of our current work eliminate the need for such possibly subject- and interface-specific preprocessing steps. Moreover, they report a comparably low average precision score of only 57%.

Virtual Task Predictors

Among the earliest examples of virtual task predictors is work by Campbell and Maglio [11]. They use a wide range of eye movement patterns in order to classify reading, skimming, and scanning tasks.

This was followed to a great extent by studies concentrating on intention prediction, i.e. predicting whether the user wants to interact with the system or not during natural interaction. For instance, Bader et al. [3] use a probabilistic model to predict whether the user intends to select a virtual object or not with 80.7% average accuracy. Similarly, Bednarik et al. [6] use SVMs to predict whether the user intends to issue a command or not with 76% average accuracy. Both prediction tasks are examples of binary classification, which indicates that in both cases, baseline accuracy score corresponding to the random classifier is 50%. This cannot be compared to our case where the baseline accuracy score is merely 20%. Hence, reported accuracy scores and classifier gains (defined as the measurement of improvement over the random classifier) should be interpreted in consideration of this fact. Moreover, as far as we know, none of these works have carried out formal studies to evaluate the proposed prediction models in online usage scenarios that involve real users interacting with predictive user interfaces driven by these models.

To our knowledge, only a few studies exist that take intention prediction one step further and attempt multi-class intention prediction of virtual tasks. The first notable example is by Courtemanche et al. [19] who claim their approach to activity

recognition to be the first one to incorporate eye movements. This work utilizes eye movements discretized in terms of interface-specific AOIs in addition to keystrokes and mouse clicks input by the user during interaction. They use HMMs to predict which of the three Google Analytics tasks (i.e. evaluating trends in a certain week, evaluating new visits, and evaluating overall traffic) the user is currently performing with 51.3% average accuracy. The second example is recent work by Steichen et al. [74]. Their domain is information visualization with graphs including bar graphs and radar graphs. Similarly, they rely on interface- and graph-specific AOIs for feature extraction and Logistic Regression to predict which of the five information visualization tasks (retrieve value, filter, compute derived value, find extremum, and sort) the user is currently performing with 63.32% average accuracy.

The superiority of our work over these two studies is threefold. First, both of these studies are interface-dependent since they analyze eye movements with respect to predefined AOIs. In contrast, our work avoids possibly subject- and interface-specific preprocessing steps common in gaze-based systems. Second, possible application areas of both of these studies are highly specific and limited since the corresponding task prediction models focus on Google Analytics tasks and graph-based information visualization tasks, respectively. On the contrary, our work can be applied in all areas that utilize basic interaction tasks like dragging, resizing, and scrolling. Accordingly, the application areas can range from simple interfaces to more complicated document or image editing software. Third, our prediction system is comparably more accurate, which makes it a better candidate for practical use.

In subsequent studies, Carenini, Conati, Steichen et al. propose different user interface adaptations for graphs (e.g. highlighting, drawing reference lines, and recommending alternative visualizations) [13], and study the effects of these adaptations on a user's performance, both in general and in relation to different visualization tasks and individual user differences [18]. However, as the authors also mention in a recent publication [75], they have still not published a fully integrated adaptive information visualization system that is able to dynamically provide adaptive interventions that

are informed by real-time user behavior data.

2.1.3 Gaze-Based Intelligent User Interfaces

To the best of our knowledge, there is no line of work that uses online feedback from a gaze-based task prediction model to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. The majority of the related work focuses solely on generating prediction models and evaluating them in terms of prediction accuracy. However, they have paid little attention to how these prediction models would perform in online usage scenarios. In this thesis, we address the multi-faceted goal of building a real-time user interface that dynamically captures and predicts user's task-related intentions and goals based on eye-movement data, and proactively adapts itself according to these predictions.

A closely-related research area focuses on *gaze-contingent user interfaces* [26]. Gaze-contingent user interfaces utilize gaze data for adapting the user interface contents as we do. However, they rely simply on the instantaneous location of a user's focus of attention. Besides, they do not contribute probabilistic prediction systems or sophisticated gaze-based feature extraction mechanisms to the literature. Nevertheless, for completeness sake, our literature review covers works in this area as well.

Although very few publications address the issue of building gaze-based predictive user interfaces, gaze-contingent user interfaces have attracted much attention from research teams in the last decade. Gaze-contingent user interfaces alter the on-screen view presented to the user based on the focus of a user's visual attention. These interfaces are utilized for improving the usability in information visualization applications, promoting engagement and learning in e-tutoring applications, etc. Despite manifesting the large potential benefits of gaze-contingent user interfaces in numerous application areas, all existing works have the following shortcomings in common. They are rule-based, i.e. they tie specific actions to specific regions on the screen and trigger the user interface for an adaptation only based on the duration of eye gaze

on these specific regions. In these works, there is no probabilistic prediction system that directs the adaptive behavior of the user interface. Accordingly, there is no effort to tackle challenges associated with uncertainty or prediction errors. There is no systematic analysis investigating whether and how these user interface adaptations affect user's natural gaze behavior. Lastly, there are very few formal studies to assess the usability and perceived task load associated with these user interfaces.

One of the first examples of gaze-contingent user interfaces is proposed by Starker and Bolt [73]. Their system uses dwell time to determine which part of a graphical interface a user is interested in, and then provides more information about this area via visual zooming and synthesized speech. Starker and Bolt [73] have notable contributions that use gaze data for adapting the contents of information visualization systems. Streit et al. [76] use gaze data to enlarge visualization or maximize clarity of focused regions in 2D scenes, and to navigate 3D scenes. Okoe et al. [56] use gaze data to improve a user's speed and accuracy in determining whether two nodes are connected in a graph by dimming out or highlighting edges according to user's view focus, and manipulating saliency of sub-graphs around nodes viewed often. Several publications have appeared in recent years documenting the use of gaze-contingent user interfaces in intelligent tutoring systems. Sibert et al. [70] use dwell time to detect difficulties in identifying words during reading tasks and assist users by providing visual (via highlighting) and auditory cues. Wang et al. [80] and D'Mello et al. [24] use gaze data to alleviate disengagement during learning by providing visual and auditory feedback to "unattentive" students looking away from the screen.

To the best of our knowledge, among the existing works that aim to build gaze-contingent user interfaces, there is no work that addresses the problem of adapting the user interface contents in line with user's task-related intentions and goals inferred via probabilistic models of user behavior.

2.2 Gaze-Based Biometric Authenticaton

In this section, we present the related work in gaze-based behavioral biometrics literature under four main sub-sections, where each sub-section corresponds to one of our contributions. Our first contribution is a robust gaze-based biometric authentication system. However, as also mentioned by Holland and Komogortsev [33], it is not yet possible to compare the robustness of existing authentication systems since each use a different set of tasks for identifying/authenticating a different number of subjects based on different feature extraction methods. More importantly, each work employs a different metric for assessing robustness. In the subsequent sub-sections, we discuss how our novel set of tasks, multimodal feature representation, and multimodal database fill the gaps in gaze-based behavioral biometrics literature.

2.2.1 Gaze-Based Biometric Authentication Tasks

The idea of using the eyes for biometric authentication has a long history. First wave of researchers pioneered by Simon and Goldstein [72] focused on using the unique patterns on a person's retina blood vessels for biometric authentication, a technique known as *retinal scanning*. Second wave of researchers pioneered by Doggart [25] and Adler [1] offered to use the complex random patterns ("minute architecture" as Doggart puts it) of a person's iris for biometric authentication, a technique known as *iris recognition*. Today, both techniques are actively used for security purposes by government agencies, airports, banks etc. However, despite their commercial success, both techniques focus on physiological characteristics, and are therefore deprived of the advantages of behavioral biometrics systems that exploit information directly originating from brain activity [65].

Using eye gaze behavior patterns for biometric authentication was first suggested by Kasprowski and Ober [39]. In this work, the subjects are presented with a single jumping point on an otherwise blank screen and their eye movement data are recorded while they follow the point on the screen. Similarly, Komogortsev et al. [43] and Liang et al. [48] employ a jumping point stimulus for gaze-based biometric authentication.

The use of a jumping point as the task stimulus is, however, problematic since the experiment interface dictates where the subjects should look at any moment, and the subjects are not "free" to decide on their own. As a result, that kind of stimulation mostly examines physical aspects of the oculomotor system rather than behavioral characteristics of the eye [39, 65]. In summary, these studies, unfortunately, fail to make use of information emerging from "the brain's decision center" as Kasprowski and Ober agree.

On the other hand, a vast majority of researchers present the subjects with a static stimulus in various forms. Prominent examples include studies that use a cross image [5, 54], a face image [12, 23, 65], a text excerpt [33], a static web page with navigational links [7], and a Rorschach inkblot [44]. In these studies, the authors ask the subjects to view the related static stimulus for a preset amount of time. However, the use of a static stimulus is problematic as well. This problem is best described with the phenomenon termed "learning effect". After repeated exposure to the identical static stimulus, subjects tend to stop eye movements on the familiar visual content [39, 48]. For example, when subjects are repeatedly given the same text excerpt, they are observed to have "lazy eyes" and skim the excerpt instead of reading it [33]. However, permanence is a key feature for biometric authentication systems, i.e. these systems are expected to be invariant over time. These studies, unfortunately, fail to meet this criterion since subjects develop a sense of familiarity with static stimuli over time, which in turn has an adverse effect on subjects' eye gaze movements and the corresponding biometric features.

Only a few researchers avoid the disadvantages of using jumping point or static stimuli for gaze-based biometric authentication, and employ more appropriate and useful tasks for this purpose. One example work is by Silver and Biggs [71], where the subjects are asked to read and type words into a computer system while their eye movement data are being recorded. Although their task of choice is familiar to users and easy/quick to perform, their gaze-based authentication model "failed to be successful" since authentication is made solely based on keystroke-based features such

as the average time to press the space bar or the total number of typing mistakes. This makes the appropriateness and usefulness of this task questionable for gaze-based biometric authentication. Another example work is by Kinnunen et al. [41], where the subjects are asked to watch a 25 minutes long video with subtitles. The authors report their results based on varying lengths of training and test data. Best results are for 9 minutes of training and 10 seconds of test data. Even if the users are convinced to watch a 9 minute video once for enrollment, asking the users to watch the same video for 10 seconds each time they want to unlock their mobile devices is perceivably unacceptable, or at least impractical. A recent and closely related work is by Darwish and Pasquier [20]. The authors use four different stimuli for extracting characteristic gaze behavior: the first two are "active tasks" that require the subjects to replicate or input a pattern on a pad whereas the last two stimuli are "passive tasks" that only ask the subjects to view a plot or an image. The best results are achieved with the active tasks, confirming (1) the disadvantages of using static stimuli for gaze-based biometric authentication, and (2) the appropriateness and usefulness of our tasks that all ask the subjects to input a memorized pattern, and therefore require "long term cognition activation". One minor problem with this work concerns task durations since the first two tasks need 17 seconds and 8 seconds for user authentication, respectively. In summary, there is no task in the literature that is both familiar to users, easy/quick to perform, and able to consistently elicit characteristic gaze behavior from users for biometric authentication.

2.2.2 Multimodal Feature Representations for Gaze-Based Biometrics Research

There exist a few studies that combine gaze-based features with other ocular features. Bednarik et al. [5] use distance between the eyes, Komogortsev et al. [44] use anatomical properties of the human eye and the physical structure of the iris, and Darwish and Pasquier [20] use iris constriction and dilation parameters for extracting additional features to be combined with the gaze-based features. However, we argue that none of these studies are truly multimodal since all features are computed based

on the physical/dynamic characteristics of the human eye itself.

Silver and Biggs [71] present the only work that truly attempts to combine gaze modality with another modality to develop a multimodal biometric authentication system with enhanced robustness. The authors propose to use keystroke-based features along with gaze-based features. However, as also mentioned in the previous sub-section, gaze-based features "failed to be successful" on their own and degraded the performance of the authentication system when combined with keystroke-based features. Therefore, this work is more a contribution to keystroke-based biometrics literature rather than to gaze-based or multimodal biometrics literatures. In summary, there is no feature representation in the literature that fuses information collected via different modalities in order to verify a user's authenticity.

2.2.3 Multimodal Databases for Gaze-Based Biometrics Research

Komogortsev et al. [44] and Cantoni et al. [12] present the only two studies that contribute a database to the gaze-based biometrics research community. Both databases consist of ocular data collected from a large set of participants while they view static images. Albeit large, both databases consist solely of ocular data, i.e. both databases are unimodal.

Chapter 3

GAZE-BASED VIRTUAL TASK PREDICTION SYSTEM

Our overall approach to gaze-based prediction of virtual interaction tasks is depicted in Fig. 3.1. The left part of the diagram shows how we build our system whereas the right part shows how our system performs predictions. After we build our system, it can be used to infer user's task-related intentions and goals in an event-driven manner where each pen marking triggers prediction.

Briefly, our system is built as follows: Initially we collect sketch and gaze data during a number of pen-based interaction tasks and build a multimodal database. We then extract novel gaze-based features from this database and train a task prediction model using supervised machine learning techniques. These steps are executed only once. Then, our system is ready for prediction. Detailed description and discussion of our approach can be found in the following sections.

We have three main contributions. First, we present a carefully compiled multimodal dataset that consists of eye gaze and pen input collected from participants completing various virtual interaction tasks. Second, for predicting user intention through gaze, we propose a novel gaze-based feature representation based on human vision, and behavioral studies. Third, we introduce a novel gaze-based task prediction system that uses this feature representation. These features are neither subject- nor interface-specific, and perform better than commonly utilized and well-established sketch recognition feature representations in the literature. We evaluate our system based on several aspects, including the prediction accuracy, scale-invariance and generalizability across scales. In addition, we run feature selection tests to evaluate the relevance and redundancy of the feature representations. Our prediction system opens the way for more natural user interface paradigms where the role of the com-

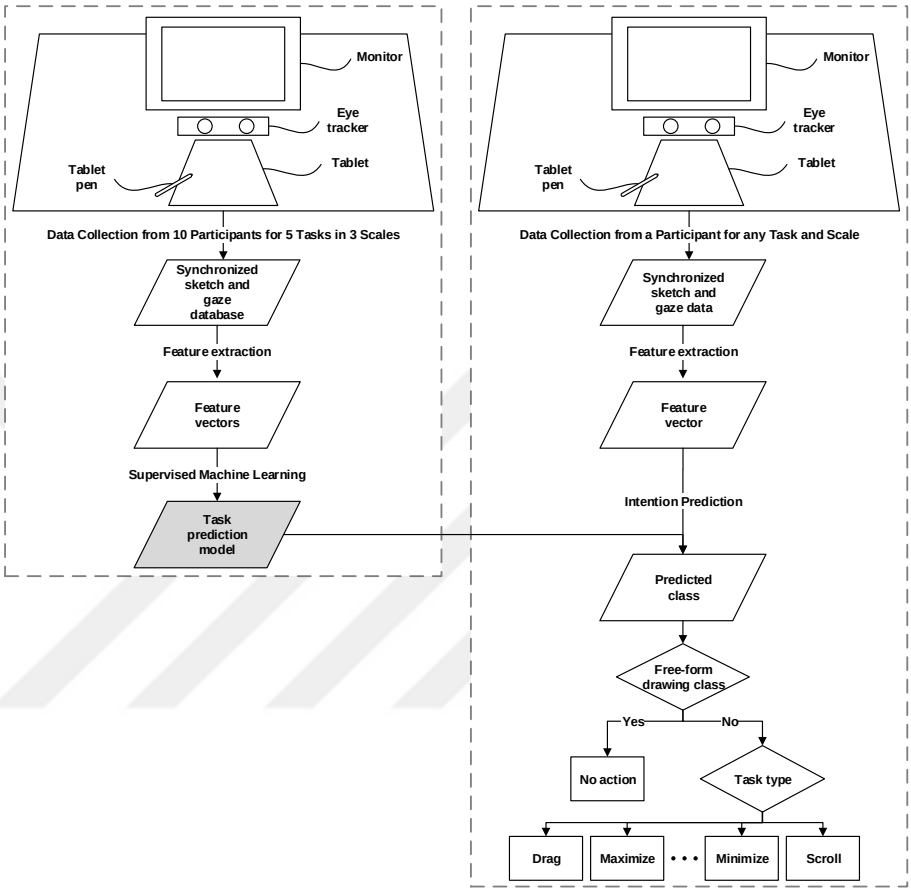


Figure 3.1: Flow diagram visualizing our overall approach to gaze-based prediction of virtual interaction tasks. The figure on the left illustrates how we build our system, and the one on the right shows how our system works in practice.

puter in supporting interaction is to "interpret user actions and [do] what it deems appropriate" [51]. It is widely accepted that intelligent mode selection mechanisms that provide low cost access to different interface operations will dominate new user interface paradigms [50].

The three major parts of our approach (i.e. data collection, feature extraction, and intention prediction) are detailed in Chapters 3.1, 3.2, and 3.3, respectively.

3.1 Multimodal Data Collection

We interpret pen and eye gaze input within a machine learning framework. This primarily requires large amounts of data for training classifiers. We collect data in a controlled setup where the users are asked to carry out a number of pen-based virtual interaction tasks.

3.1.1 Physical Setup

To create a database composed of synchronized sketch and gaze data, we use a tablet and a Tobii X120 stand-alone eye tracker for the sketch and gaze modalities, respectively. The eye tracker needs to be calibrated once for each user. The calibration step is brief, and it is posed as an "attention test" to conceal any hints of eye tracking from the user. Tobii X120 operates with a data rate of 120 Hz, tracking accuracy of 0.5° and drift of less than 0.3°. The tracker allows free head movement inside a virtual box with dimensions 30×22×30 cm.

The physical setup for data collection is depicted in Fig. 3.2. Note that the drawing surface and the display are separated. In particular, the drawing surface (i.e. a tablet) is placed below the eye tracker and the eye tracker is placed below a monitor. This physical configuration allows us to collect pen input given the technical limitations of the Tobii X120 eye tracker. More specifically, the general setup guidelines for the eye tracker require placing it below the interaction screen. However, placing the eye tracker below the tablet inevitably leads to user's arm blocking the eye tracker's field of operation. To overcome this problem, the drawing surface and the display are separated in our setup. To facilitate hand-eye coordination during interaction, we render a pen-shaped visual cursor on the display indicating the position of the user's pen on the tablet.

3.1.2 Data Collection Tasks

Our data collection process is designed to include frequently employed pen-based virtual interaction tasks. These tasks, depicted in Fig. 4.3, are: *drag*, *maximize*, *mini-*

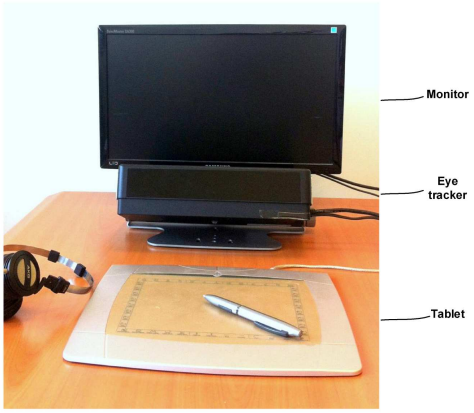
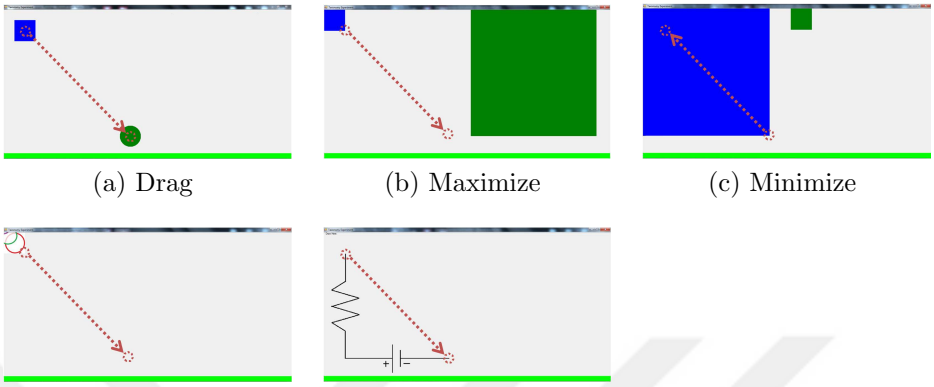


Figure 3.2: Physical setup for multimodal data collection. Input and display are separated resulting in an indirect input configuration [29].

mize, *scroll*, and *free-form drawing*. Typical pen-based interaction consists of *stylized* and *non-stylized pen inputs*. Stylized pen inputs consist of symbols and gestures which have characteristic visual appearances (Fig. 3.4). Hence, they can be classified with conventional image-based recognition algorithms. On the other hand, non-stylized pen inputs lack a characteristic visual appearance, and appearance alone does not carry sufficient information for classification purposes. Therefore, in order to test out our system's prediction power in a more challenging setting, we selected tasks that yield non-stylized pen input. In particular, for each task the stylus has an approximate starting point and an approximate ending point. In order to complete each task, the user needs to make a movement that starts near the starting point and ends near the ending point. Pen input corresponding to these tasks do not have characteristic visual appearances and do not lend themselves well to conventional image-based recognition algorithms. Instructions given to the users for each task are summarized in Table 3.1[2].

In addition to being frequently employed, these tasks also have the following properties in common:

[2]Note that the instructions for the *drag*, *maximize*, and *minimize* tasks contain color information which will not show in a B/W copy of Fig. 4.3. For these tasks, the object to be manipulated (dragged/maximized/minimized) is the one on the left side of each screen.



(a) Drag (b) Maximize (c) Minimize

(d) Scroll (e) Free-Form Drawing

Figure 3.3: Pen-based virtual interaction tasks included in our research. Starting and ending regions of desired pen motion in each task are visualized with dotted circles. In the rest of the thesis, the center points of these regions will be referred to as *anchor points*. Direction of the desired pen motion in each task is visualized with a dotted arrow connecting the starting and ending regions. It is important to note that the dotted visualizations only serve as a reference within this document and they are not shown to the user during data collection.

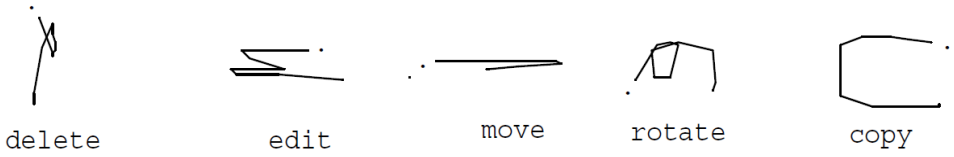


Figure 3.4: Common editing gestures [67] serve as examples of stylized pen input. Each gesture has an easily distinguishable characteristic visual appearance.

- Each task can be carried out using a tablet and a stylus.
- Each task necessitates continued visual attention for planning and guiding the hand/eye movements. Thus during each task, user's eyes are expected to remain on the display device.
- Tasks last roughly the same amount of time.

The *free-form drawing* task differs from the remaining tasks in a special way. Unlike the other pen-based virtual interaction tasks, this task is included in our study in an attempt to model and avoid unsolicited task activation. More specifically, since

Task	**Instruction**
Drag	Drag the blue square onto the center of the green circle.
Maximize	Increase the size of the blue square to match the size of the green square.
Minimize	Decrease the size of the blue square to match the size of the green square.
Scroll	Pull the chain until the color of the last link is clearly visible.
Free-form Drawing	Connect the battery and the resistor with a wire.

Table 3.1: Instructions given to the users for each task during data collection.

our prediction system is to be employed in a proactive user interface, the ability to distinguish between the intention to sketch and the intention to interact becomes vital. Accordingly, the *free-form drawing* task is included in our study to distinguish pen movements that are intended to activate the proactive user interface for task execution. Otherwise, the user would find that all pen movements (intended or not) activate a new task execution.

3.1.3 Data Collection Interface

To collect multimodal data for the mentioned tasks, we designed and implemented a custom application. We collected sketch data using the Microsoft Managed INK Collection API (Pen API) and INK Data Management API (Ink API). These APIs capture pen coordinates online, and save digital ink packets captured between pen-down and pen-up events as strokes. We collected gaze data using the Tobii Analytics Software Development Kit (SDK). The collected gaze data is composed of gaze points, each represented as an array of tuples consisting of local UNIX timestamp, remote UNIX timestamp, validity code, horizontal location, and vertical location information sampled at 120 Hz. Validity code, horizontal location, and vertical location information are obtained for the left and right eyes individually.

Our user interface has the following properties:

- Sketch and gaze data are collected in a time-synchronized fashion.
- A gaze tracking status bar visualizes whether the eye tracker is able to find both eyes. Users can monitor and adjust their posture based on the gaze tracking status bar. The bar stays green as long as the eye tracker is functioning properly, but turns red if the eye tracker loses the eyes. Gaze data packets collected while the status bar is red are marked as *invalid* by the eye tracker. The status bar is disguised under the name of a *posture check indicator* in an attempt to avoid any hints of eye tracking.
- At the beginning of each task, prerecorded non-distracting (in terms of avoiding unsolicited gaze behavior) audio instructions are delivered via headphones.
- After the completion of each task, the percentage of *valid* gaze data is calculated to assure that at least 80% of the collected gaze data is *valid*. In cases where fewer than 80% of the gaze packets are valid, the current task is automatically repeated and the user is warned via an audio message instructing him/her to assume a correct posture and maintain an appropriate distance to the monitor.

When users execute a task, positions of the pen-down and pen-up events respectively define the starting and ending points of the task. To insure that starting and ending points of a task do not act as confounding variables in our data collection process, the tasks were designed to have coincident starting/ending points (Fig. 4.3).

3.1.4 Database

We refer to each run of a certain task at a certain scale as *a task instance*. Our multimodal database consists of 1500 task instances collected from 10 participants (6 males, 4 females) over 10 randomized repeats of 5 tasks across 3 scales. The participants were recruited from undergraduate and graduate students of Koç University's Faculty of Engineering on a voluntary basis.

Multimodal data was collected across three different scales to test our system in terms of scale-invariance and generalizability across scales. The scale variable determines the length of the path connecting the two anchor points present in each

task (Fig. 4.3). These three scales will be referred to as *large*, *medium*, and *small*, respectively. Lengths of the paths corresponding to each scale were set in light of facts about human vision. The spatial field of vision is functionally divided into three regions, foveal, para-foveal, and peripheral. As summarized in Table 3.2, each region has distinct characteristics with respect to acuity limitations; therefore, lengths of the paths were set to 21 cm, 10.5 cm, and 5.25 cm for the *large*, *medium*, and *small* scales, respectively.

Type of region	Foveal	Para-foveal	Peripheral
Limit (in degrees)	<2	2 – 10	>10
Limit (in cm)*	<2.44	2.44 – 12.25	>12.25

*Calculated based on acuity limit of each type of region in degrees and distance of the user to the monitor which is 70 cm in our setup.

Table 3.2: Different regions of human spatial field of vision [63].

At the beginning of each data collection process, users were presented with 10 practice runs consisting of one run in *large* scale and one run in *small* scale for each of the 5 tasks.

Sketch data timestamps of two task instances were missing; thus, those instances were omitted from the database. In addition, invalid gaze data was filtered out using the validity codes[3].

3.2 Novel Gaze-Based Feature Representation

Our system utilizes only two kinds of features for gaze-based task prediction: *Instantaneous Distance Between Sketch and Gaze Positions* and *Within-Cluster Variance of Gaze Positions*. The strength of these features stems from the fact that they eliminate the need for possibly subject- and interface-specific preprocessing steps common in gaze-based systems. Some examples of these common error-prone preprocessing

[3]Validity code information is an estimate of how certain the eye tracker is able to correctly identify both eyes. Validity codes can take on a set of specific values. The Tobii SDK Developer's Guide recommends all samples with validity codes 2 or higher to be discarded.

steps include segmentation of gaze data into fixations and saccades and manual specification of regions of interest. Below, we describe each feature in detail, as well as the rationale behind how they are expected to aid task identification.

3.2.1 Feature 1: Instantaneous Distance Between Sketch and Gaze Positions

Let $G_t < x, y >$ be the x and y positions of the gaze on the screen at time t during the execution of a particular task. Let $P_t < x, y >$ represent the position of the tip of the stylus at time t. We argue that the distance between these points $D_t = |G_t - P_t|$ evolves in a strongly task-dependent fashion throughout the completion of a task instance. In other words, distance curves D_t computed for task instances of the same type have similar rise/fall characteristics, while those of different task types have quite different profiles. Unfortunately, even for the same task, the distance curves will evolve at different rates, hence they will not be identical. Assuming that we could compute representative characteristic curves for all task types, we could then compare the distance curve of an unknown task instance to these characteristic curves, and use the degree of matching as a useful feature for task identification. Below we describe the rationale behind the instantaneous distance feature, and suggest a method for computing task-specific characteristic curves.

The Rationale

Hand-eye coordination behavior inherent in virtual interaction tasks changes over the course of a task instance as a function of changes in user sub-tasks [4, 27, 32, 37]. The multiple steps of each task can be thought of as consecutive sub-tasks and each sub-task entails a different type of hand-eye coordination behavior. The rationale behind the first feature of our novel gaze-based feature representation is based on this observation and attempts to capture the goal-dependent dynamic aspects of human hand-eye coordination behavior through the evolution of the distance between instantaneous gaze and sketch positions calculated over a task instance.

Consider the task in Fig. 4.3b. In a typical instance of this task, the user is

instructed to drag a source object (the blue square) onto a target object (the green circle). The sub-tasks of this task are (1) positioning the pen on the source object, (2) determining the position of the target object, and (3) dragging the source object towards the target object. We argue that the dynamic aspects of human hand-eye coordination behavior are reflected in the distance values between instantaneous gaze and sketch positions calculated over time. Fig. 3.5 and Fig. 3.6 generated from the same sample task instance support our argument. Fig. 3.5 gives a visualization of the user's sketch data along with a number of sketch and gaze data samples. Sketch and gaze data points collected at identical time instances are connected with dotted lines. The length of a connection line denotes the value of the sketch-gaze distance feature for the corresponding instance. Fig. 3.6 demonstrates how the value of this feature changes over time. In this figure, the sketch-gaze distance feature is plotted for the same user and same task, over three different scales. Peaks of the plots could conceivably mark the second sub-task during which the user, after having positioned the pen on the source object, is now gazing at the target object. Note that sketch-gaze distance feature expresses similar characteristics across different scales; thus our approach and our novel feature can be generalized and applied to data collected across different scales.

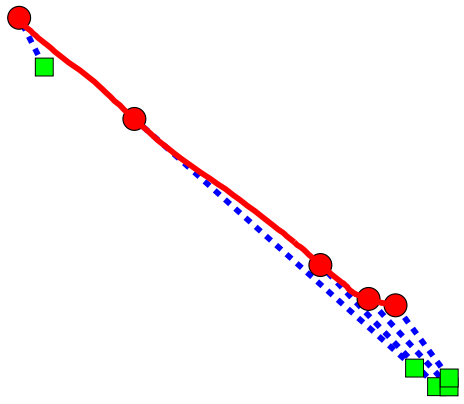


Figure 3.5: Visualization of the user's sketch data (solid line) along with a number of sketch (circles) and gaze (squares) data samples. Dotted lines connect the instantaneous sketch and gaze sample pairs.

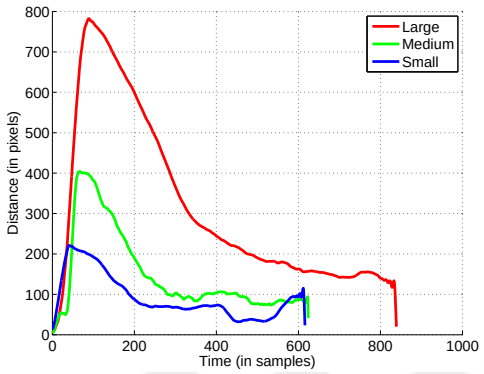


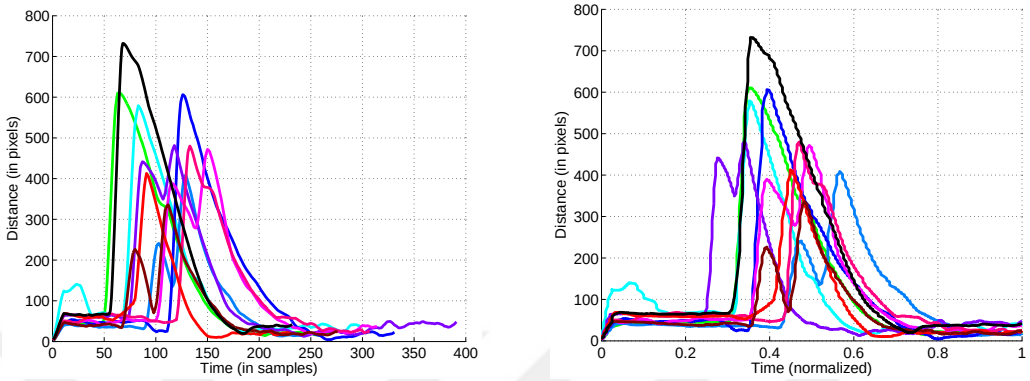
Figure 3.6: Visualization of the changes in the value of sketch-gaze distance feature as a function of time.

Inspection of the sketch-gaze distance curves for the *drag* task reveals that the rapid rise and gradual decline behavior is typical of all instances of the *drag* task. Similarly, the distance curves for the other tasks also display task-specific characteristic rise and fall behaviors. We compute distinct sketch-gaze distance curves for each virtual interaction task using sketch-gaze distance curves of all task instances. These task-specific sketch-gaze distance curves will be referred to as *characteristic curves*.

Characteristic Curve Extraction

Fig. 3.7a illustrates the sketch-gaze distance curves corresponding to 10 repeated task instances of a user for the *drag* task in the large scale. These curves have been filtered by a symmetric Gaussian low-pass filter of size 11×1 and $\sigma = 5$. It is evident that the user naturally spent different amounts of time to complete each task instance. In order to overcome the discrepancy in task completion times, sketch-gaze distance curves are normalized with respect to a standard time range as depicted in Fig. 3.7b. However, even after the normalization procedure, the sketch-gaze distance curves are still not sufficiently aligned. This indicates that although users accomplish similar sub-tasks to complete each task, the completion time and speed of these sub-tasks vary across task instances, and even within a user.

To align sketch-gaze distance curves that are similar in shape but evolve at dif-



(a) Original sketch-gaze distance curves (b) Normalized sketch-gaze distance curves

Figure 3.7: Sketch-gaze distance curves corresponding to 10 repeated task instances of a user each drawn in a distinguishing color.

ferent rates, we use *dynamic time warping* [68]. Dynamic time warping is a sequence alignment method often used in the time series classification domain to measure the similarity between two sequences independent of non-linear variations in the time dimension. We use it both for computing the similarity of two given curves and for finding an optimal alignment between them. Fig. 3.8 demonstrates the alignment of two curves using dynamic time warping[4]. Alternative sequence alignment methods include, but are not limited to, functional data analysis [62], curve alignment by moments [36], and curve synchronization based on structural characteristics [42].

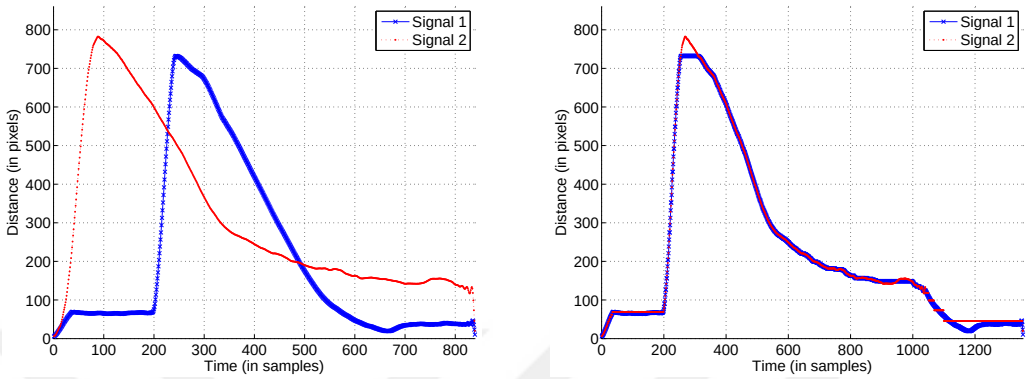
We build scale- and task-specific characteristic curves as follows:

1. For each task instance, we obtain the instantaneous sketch-gaze distance curve D_i.

2. We smooth out all D_i by a rotationally symmetric Gaussian low-pass filter of size 11×1 and $\sigma = 5$.

3. We form a similarity matrix S based on the similarity values corresponding to all possible pair combinations of sketch-gaze distance curves. Similarity values

[4]We use an open-source dynamic time warping library for MATLAB [28].



(a) Original sketch-gaze distance curves (b) Warped sketch-gaze distance curves

Figure 3.8: Sketch-gaze distance curves corresponding to two task instances of a user. We use dynamic time warping for computing an optimal alignment between two given curves by warping each curve with respect to the other one.

are computed using the dynamic time warping algorithm.

4. We create a hierarchical cluster tree from the similarity matrix. Clusters are computed using the unweighted pair group method with arithmetic mean (UPGMA) based on the Euclidean distance metric.

5. On each cluster, an algorithm we call *weighted hierarchical time warping* is applied. We developed this algorithm for computing the characteristic curve that best represents any given cluster of curves. The weight of an input curve depends on the number of leaves below the node corresponding to the input curve in the hierarchical cluster dendrogram. This way, all members of a cluster contribute equally to the resulting characteristic curve. The details of this algorithm are described in Fig. 3.9.

6. We take the final weighted hierarchical warping result of the cluster with the maximum number of elements as the characteristic curve of the respective task and scale. If there are multiple clusters containing at least 10 task instances, then each of these clusters contributes a characteristic curve to the set of characteristic curves for the respective task and scale.

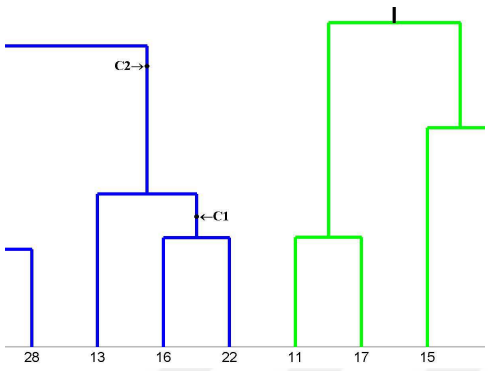


Figure 3.9: Weighted hierarchical time warping algorithm. According to this algorithm, the curve labeled *C1* is created by warping the curves with indices 16 and 22 whereas the curve labeled *C2* is created by warping the curve with index 13 and the previously created *C1* curve. Here, $\mathrm{C1} = \frac{1}{2}\times dtw(16, 22) + \frac{1}{2}\times dtw(22, 16)$ whereas $\mathrm{C2} = \frac{1}{3}\times dtw(13, \mathrm{C1}) + \frac{2}{3}\times dtw(\mathrm{C1}, 13)$. Note that *dtw(source, target)* is the dynamic time warping function that warps the source curve with respect to the target curve and returns the warped source curve. The weights determine how much the warped source curve contributes to the final warping result.

Using this algorithm, we obtain a characteristic curve for each task and scale as depicted in Fig. 3.10 for the large scale. Pseudocodes of our algorithm can be found in Appendix B. Given a sketch-gaze distance curve, we construct its feature vector by measuring its similarity to each of these characteristic curves. This vector corresponds to the first feature of our novel gaze-based feature representation. Again, similarity values are calculated using dynamic time warping. Although not shown in this figure, some tasks may have multiple characteristic curves. This happens if there exist multiple strategies that users follow to complete a specific task. Therefore, the length of this feature vector is not constant and depends on the total number of characteristic curves.

A qualitative investigation of the characteristic curves brings up interesting observations on stylus-gaze coordination behavior. In line with our initial argument, this behavior is observed to be task-dependent. Furthermore, the characteristic curves have easily interpretable shapes. For instance, in the *scroll* task, the hand keeps pulling the chain while the eyes are busy attending to the newly appearing informa-

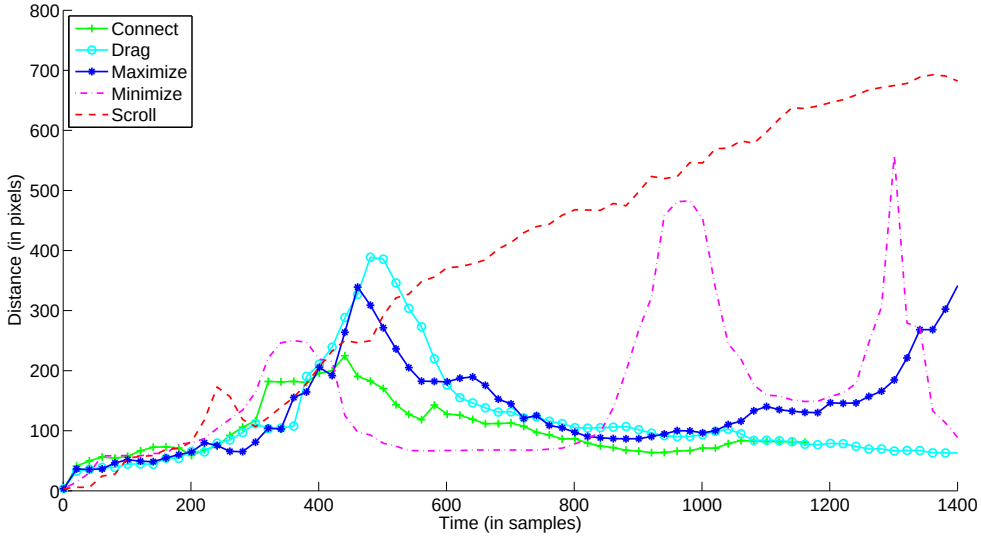


Figure 3.10: Characteristic curves obtained from sketch-gaze distance curves of each task in large scale.

tion on the display. Therefore, one would expect the distance between the hand and the eyes to increase constantly; our findings agree with this expectation (Fig. 3.10).

3.2.2 Feature 2: Within-Cluster Variance of Gaze Positions

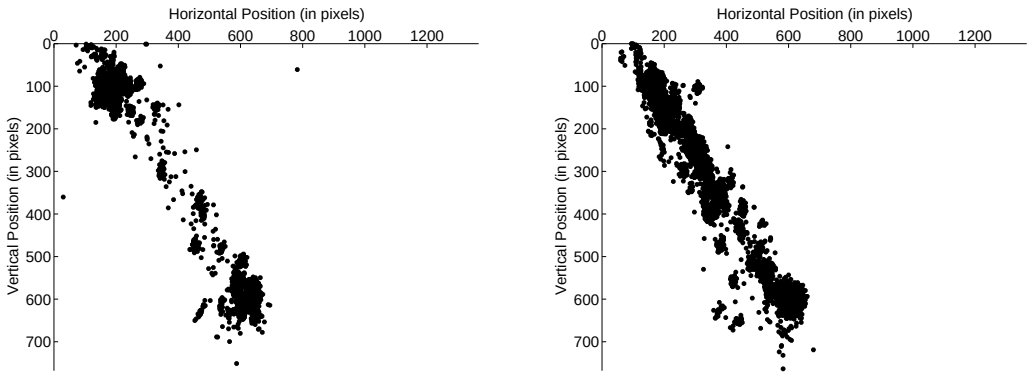
As we demonstrate in the next subsection, eye gaze positions along the path of different virtual interaction tasks exhibit different clustering behaviors. Hence, a measure of how the gaze points are clustered and spread out along the trajectory of the task carries discriminative information for task identification. This is what we attempt to capture with the within-cluster variance feature.

The Rationale

Humans employ two different modes of voluntary gaze-shifting mechanism to orient the visual axis. These modes are referred to as saccadic and smooth pursuit eye movements. It is widely accepted that "saccades are primarily directed toward stationary targets whereas smooth pursuit is elicited to track moving targets" [21]. Typical vir-

tual interaction tasks contain both stationary and moving targets. A user's attention can be dominantly directed towards targets of either type depending on the intended task.

Our experiments show that in a typical *drag* task, saccades are more common and the user's attention is drawn from one stationary target which is the initial position of the object currently being dragged to the other stationary target which is the intended position of the object (Fig. 3.11a). Conversely during *free-form drawing* (Fig. 3.11b), smooth pursuit is more common and the user's attention is drawn to the moving target (the newly appearing ink). In saccades, gaze points accumulate around the stationary targets whereas in smooth pursuit, gaze points scatter along the pursuit path. The second feature of our novel gaze-based feature representation is based on these observations, and hence attempts to quantify how the eye gaze data is structured in terms of saccades and fixations.



(a) Gaze data for the *drag* task. Saccadic eye movements result in gaze point clusters with low within-cluster variance.

(b) Gaze data for the *free-form drawing* task. Smooth pursuit eye movements result in gaze point clusters with high within-cluster variance.

Figure 3.11: Gaze data corresponding to 10 repeated task instances of a user.

Quantifying the Distribution of Saccades and Fixations

We quantify the distribution of saccades and fixations by measuring the mean within-cluster variance of clustered gaze points for each task instance. Clustering is done

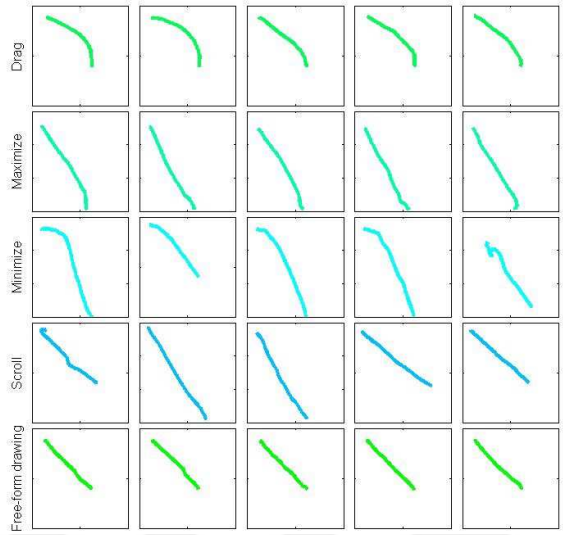


Figure 3.12: Sketch data corresponding to a user's 5 repeated task instances for 5 tasks. Pen trajectories for our tasks serve as an example for non-stylized pen input that do not have easily distinguishable characteristic visual appearance.

via MATLAB's *k-means clustering* algorithm and repeated three times for different k values as $k = 1, 2, 3$. Thus, the length of this feature vector is constant and equal to 3. We do not use higher orders of k since gaze packets aimed at the source and target objects respectively form the first and second clusters while the remaining gaze packets form the third cluster.

3.3 Intention Prediction and Evaluation

In this section, we evaluate the effectiveness of the features introduced above in predicting virtual interaction tasks. During evaluation, we focus on several aspects, including the prediction accuracy, scale-invariance and generalizability across scales. In addition, we run feature selection tests to evaluate the relevance and redundancy of the features introduced above. We also compare the prediction power of our novel gaze-based feature representation to that of commonly utilized and well-established sketch-based feature representations in the literature.

As mentioned earlier in Chapter 3.1, we record participants' eye gaze as well as

the pen trajectory during the execution of each task instance. A subset of sketch data from our database is shown in Fig. 3.12. As seen in Fig. 3.12, even though the individual pen trajectories for our tasks do not appear to be as stylized as those in Fig. 3.4, it is still conceivable that pen trajectories alone may suffice for accurate task prediction. To this end, we experimented with a number of image-based approaches to extract features from the collected sketch data. These feature representations, IDM Features [57] and Zernike Moments [40], are shown to work well for hand-drawn sketch data by [78]. The authors further demonstrate that to achieve good recognition accuracies with these feature representations, good feature extraction parameters must be selected. IDM Features have three free feature extraction parameters as k (kernel size), σ (smoothing factor), and r (resampling parameter). Zernike Moments have one free parameter, which is the order of the Zernike moment o. We set the parameters of the evaluated sketch-based feature representations in accordance with the optimum values reported in [78]. For reproducibility, our parameter settings are listed in Table 3.3.

Feature representation	**Parameter settings**
IDM features	$k = 25$, $\sigma = 3$, and $r = 150$
Zernike moments	$o = 12$

Table 3.3: Parameter settings for the sketch-based feature representations.

All accuracy tests were done using the LIBSVM [14] implementation of Support Vector Machines. The accuracies are measured in line with the standard three-step machine learning pipeline, where we first extract feature vectors from a set of data samples, then train classifier models using these feature vectors, and finally measure accuracies using unseen data.

1. We partition the input data into 5 disjoint folds, chosen randomly but with roughly equal size. Out of these 5 folds, 4 are reserved for training and validating the model whereas the remaining fold is reserved for testing the model.

2. We extract feature vectors from the training data, and normalize them by standardization.

3. We train a model using the Gaussian radial basis function (RBF) kernel. We estimate the hyper-parameters of our model using grid search with 5-fold cross-validation.

4. We evaluate the resulting prediction model on the testing data to obtain accuracy.

5. Steps 2-4 are repeated for each random split in a round-robin fashion such that each of the 5 folds is used exactly once for testing.

6. The mean accuracy for the random splits is reported.

3.3.1 Accuracy Tests

Our accuracy tests fall under two categories: The first set of tests evaluates gaze-based and sketch-based feature representations individually, and second set evaluates their combinations. The accuracy tests are carried out and reported for the *large*, *medium* and *small* scales, as well as for the *all scales* case, which corresponds to the entire database.

Collectively, the results of the individual tests suggest that feature representation has an effect on prediction accuracy. Specifically, our results suggest that the gaze-based feature representation is significantly better in capturing the richness, complexity, and subtlety of our user input when compared to various sketch-based feature representations that have been shown to work well for hand-drawn sketch data. On the other hand, results of the combined tests indicate that combining gaze-based and sketch-based feature representations may yield higher accuracy scores depending on the choice of sketch-based feature representation and the combination technique.

Accuracy Tests for Evaluating the Feature Representations Individually

Fig. 3.13 shows the mean accuracies for individual feature representations. We conducted a two-way ANOVA to examine the effect of feature representation and scale on prediction accuracy. ANOVA revealed a main effect of feature representation on prediction accuracy across the *Gaze-Based Features* (78.76 ± 3.84), *IDM Features* (60.46 ± 6.86), and *Zernike Moments* (38.73 ± 4.59) conditions at the $p < .05$ level, $[F(2, 48) = 292.924, p < 0.001]$. Post-hoc comparisons using the Tukey HSD test indicated that the mean score for the *Gaze-Based Features* condition was significantly higher than the *IDM Features* condition ($p < 0.001$) and the *Zernike Moments* condition ($p < 0.001$). In addition, the mean score for the *IDM Features* condition was found to be significantly higher than the *Zernike Moments* condition ($p < 0.001$).

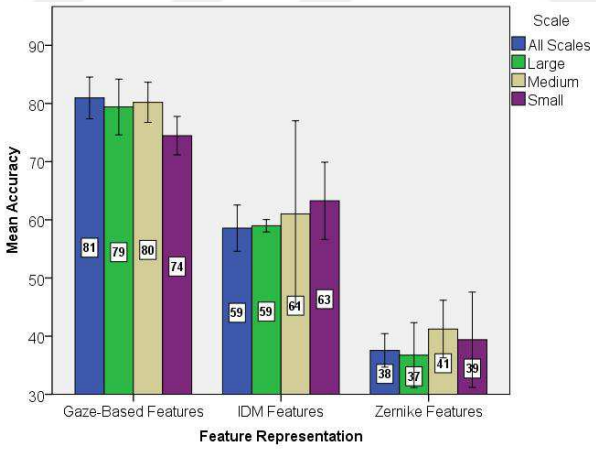


Figure 3.13: Mean accuracy scores for each feature representation and scale. Error bars indicate 95% confidence interval.

There was no main effect of scale on prediction accuracy across the *large* (58.37 ± 18.32), *medium* (60.82 ± 18.04), *small* (59.04 ± 15.88), and *all scales* (59.03 ± 18.53) conditions at the $p < .05$ level, $[F(3, 48) = 0.602, p = 0.617]$. This indicates that there is not enough evidence to show that our prediction system has a significantly higher/lower accuracy score for any particular scale irrespective of feature representation. Furthermore, there was no significant interaction between feature representation

and scale, $[F(6, 48) = 1.268, p = 0.290]$. In other words, we can infer that there is not enough evidence to show that a particular feature representation performs significantly better or worse under scale variations. Fig. 3.14 provides a graphical illustration of the interactions.

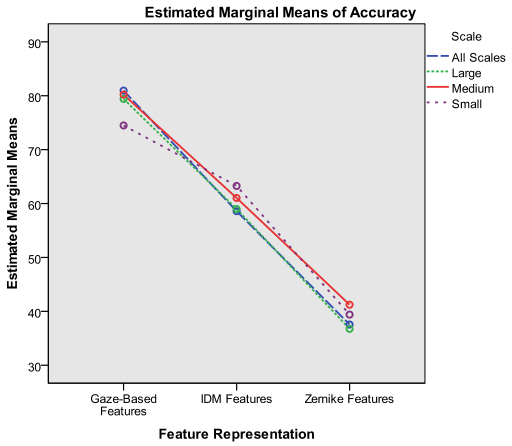


Figure 3.14: Two-way ANOVA results that examine the interaction of feature representation and scale factors on prediction accuracy.

Accuracy Tests for Evaluating Combinations of Feature Representations

The individual accuracy tests focus on the performance of individual feature representations. A natural follow-up to the previous experiments is to explore whether gaze-based and sketch-based feature representations can be combined to increase prediction accuracy. There are two common techniques for information fusion, namely classifier-level fusion and feature-level fusion. Mean accuracy values computed for each scale with all possible classifier-level fusion and feature-level fusion combinations of the gaze-based and sketch-based feature representations are shown in Fig. 3.15.

Classifier-Level Fusion: For classifier-level fusion, we train two probabilistic SVM models - one with the gaze-based features and another with either of the sketch-based features (IDM Features or Zernike Moments). The output of each probabilistic SVM model is a vector of size 5, each element of the vector representing the probability

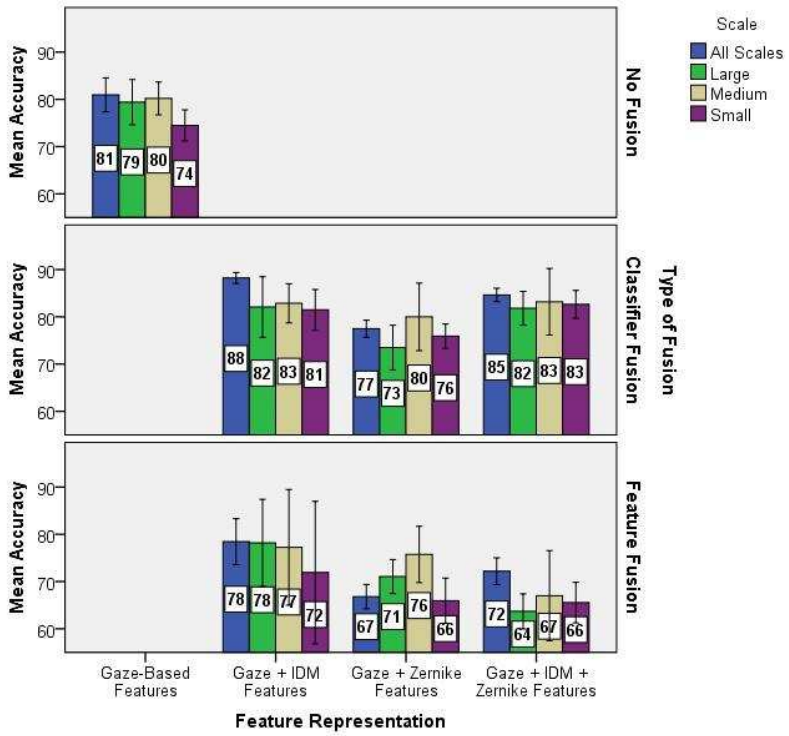


Figure 3.15: Accuracy tests with the classifier-level fusion and feature-level fusion techniques. Error bars indicate 95% confidence interval.

estimate of the input sample being a member of the five respective virtual task classes. We then use the outputs of these two probabilistic SVM models to train a third multi-class SVM model. The feature vector in this case is a vector of size 10 consisting of probability estimate values from the gaze-based and sketch-based probabilistic SVM models, respectively.

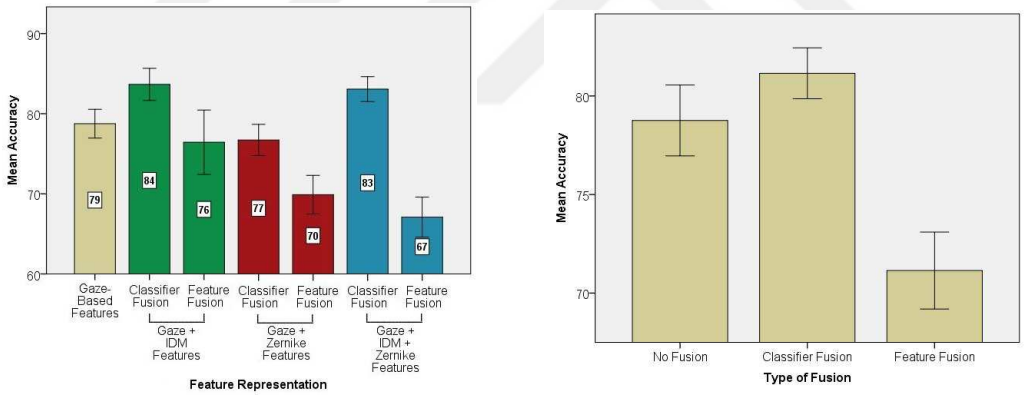
Feature-Level Fusion: For feature-level fusion, feature vectors corresponding to multiple feature representations are concatenated to construct a high-dimensional feature vector. Then, a regular SVM model is trained with this feature vector.

Statistical analysis of the accuracy tests with the combination of gaze-based and sketch-based feature representations imply the following results (For brevity, we take $p = 0.05$ unless otherwise noted.):

- Classifier-level fusion of Gaze-Based Features and IDM Features (83.66 ± 4.28)

gives the overall highest mean accuracy value (see Fig. 3.16a).

- IDM Features give higher mean accuracy values than Zernike Moments in both classifier- (83.66 ± 4.28 vs. 76.72 ± 4.16) and feature-level fusion (76.44 ± 8.59 vs. 69.89 ± 5.16) cases (see Fig. 3.16a).
- Classifier-level fusion (81.15 ± 5.01) yields higher mean accuracy values compared to feature-level fusion (71.14 ± 7.55) (see Fig. 3.16b).
- Gaze-Based Features alone (i.e. no fusion) (78.76 ± 3.84) give higher mean accuracy values compared to feature-level fusion technique (71.14 ± 7.55) (see Fig. 3.16b).



(a) Mean accuracy values for various combinations of feature representations and information fusion techniques.

(b) Mean accuracy values for the no fusion, classifier-level fusion and feature-level fusion cases. No fusion case corresponds to using Gaze-Based Features alone.

Figure 3.16: Summary of results for the combined accuracy tests. Error bars indicate 95% confidence interval.

3.3.2 Feature Selection Tests for Evaluating the Relevance and Redundancy of the Feature Representations

Our accuracy tests show that combining the Gaze-Based Features and IDM Features by classifier-level fusion gives the overall highest mean accuracy value. However, in practice, it might not be feasible to extract hundreds of features in real-time. In that

case, we can use feature selection to obtain a faster and cost-effective predictor by ranking the features based on a mutual information criterion and selecting a feasibly smaller subset of the highly ranked features. This subset is composed of the maximally relevant and minimally redundant (i.e. the best performing) features selected among all feature representations in consideration.

Feature selection tests were conducted within the Maximum Relevance & Minimum Redundancy (mRMR) feature selection framework [58]. This framework allows us to select the k maximally relevant and minimally redundant features from a total set of K features where $k \leq K$. In our case, respective lengths of feature vectors generated by Gaze-Based Features, IDM Features and Zernike Features are $f_1 = 13$, $f_2 = 720$, and $f_3 = 47$. Therefore the total number of features is $K = f_1 + f_2 + f_3 = 780$. Fig. 3.17 shows the percentage of features contributed by each feature representation to the best performing set of features. As seen here, Gaze-Based Features surpass (or equal) both sketch-based feature representations in terms of the percentage of contributed features for all values of k. All features generated by the gaze-based feature representation make their way into the best performing set of features by $k = 49$. At this point, only as little as 6.28% of the total number of features are used. Therefore, in cases where speed and cost are of concern, Gaze-Based Features offer a better alternative to IDM Features and Zernike Features.

3.3.3 Scale-Invariance Tests

Practical usage of our prediction system may involve a range of display devices and user interfaces with varying sizes and constraints. Robustness of a feature representation to variations in scale is important if we want our prediction system to work equally accurately despite these variations. Fig. 3.13 shows the mean accuracies for individual feature representations in different scales. We previously referred to this figure in Chapter 3.3.1 for our accuracy tests, but we did not focus on the scale-invariance of our task prediction system. As we substantiate in detail in the next two subsections, our task prediction system is scale-invariant in all Gaze-Based Features,

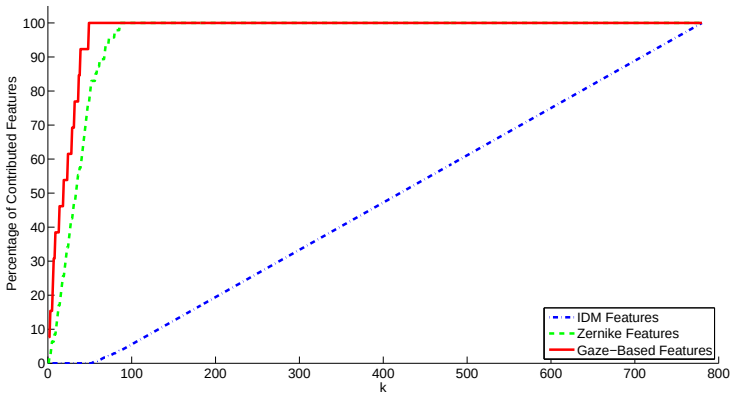


Figure 3.17: Percentage of contributed features by each feature representation to the best performing set of features selected by the mRMR framework.

IDM Features, and Zernike Features cases. The only exception is for pen and eye movements that are entirely in the foveal area. In that case, prediction accuracy deteriorates by a small, yet statistically significant amount for the gaze-based feature representation. This is expected due to limitations of our eye tracker in smaller scales in terms of tracking accuracy.

Scale-Invariance Tests with the Gaze-Based Feature Representation

In order to compare the effect of scale on prediction accuracy across the *large* (79.40 ± 3.85), *medium* (80.20 ± 2.79), *small* (74.47 ± 2.66), and *all scales* (80.95 ± 2.89) conditions, we conducted a one-way between subjects ANOVA with the gaze-based feature representation. There was a significant effect of scale on prediction accuracy at the $p < .05$ level for the four conditions $[F(3, 16) = 4.497, p = 0.018]$. Post-hoc comparisons using the Tukey HSD test indicate that the mean score for the *small* condition is significantly lower than the *all scales* condition ($p = 0.020$) and the *medium* condition ($p = 0.043$). However, there were no differences between the *all scales*, *large*, and *medium* conditions. More specifically, $p = 0.855$ for *all scales* and *large* conditions, $p = 0.980$ for *all scales* and *medium* conditions, and finally $p = 0.976$ for *large* and *medium* conditions.

Collectively, these results suggest that length of the task trajectory has an effect on prediction accuracy. Specifically, our results indicate that when the range of pen and eye movements in a task approaches the range of foveal human vision, prediction accuracy deteriorates slightly. This is expected, because tracking error is relatively worse for smaller scales. Wider ranges of pen and eye movements do not appear to increase or decrease prediction accuracy significantly. On the other hand, the most realistic test condition corresponds to the *all scales* case since data collected during natural interaction with a user interface will typically consist of tasks across a variety scales. Prediction accuracy in fact peaks at the *all scales* case.

Scale-Invariance Tests with the Sketch-Based Feature Representations

In order to compare the effect of scale on prediction accuracy across the *large* ($58.98 \pm 0.85/36.74 \pm 4.51$), *medium* ($61.02 \pm 12.90/41.23 \pm 3.99$), *small* ($63.27 \pm 5.35/39.39 \pm 6.60$) and *all scales* ($58.57 \pm 3.20/37.55 \pm 2.31$) conditions, we conducted a one-way between subjects ANOVA with IDM Features/Zernike Moments. For both feature representations, there was no significant effect of scale on prediction accuracy at the $p < .05$ level for the four conditions, more specifically $[F(3, 16) = 0.451, p = 0.720]$ for IDM Features and $[F(3, 16) = 0.942, p = 0.444]$ for Zernike Moments.

3.3.4 Tests for Assessing Generalizability Across Scales

In Chapter 3.3.3, we have presented the results of the tests on whether and how the accuracy of our prediction system varies with varying scale conditions. In all these tests, task scales of the testing data match scales of the training data. Alternatively, we can question what happens in case of mismatch. This may occur if the physical characteristics of the use case scenario differs from that of the platform used for collecting training data. For instance, we might want to deploy a prediction model trained with data collected via a large tablet display on a smart phone with a relatively smaller display. We conducted additional tests to evaluate the generalizability of our prediction model across scales with respect to the mismatch in task scales when

combined with the gaze-based and sketch-based feature representations. We tested our prediction system against six different mismatch scenarios as shown in Table 3.4.

Mismatch Type	Training Data Scale(s)	Testing Data Scale(s)	Accuracy (Gaze-Based Features)	Accuracy (IDM Features)	Accuracy (Zernike Moments)
A	Small	Medium & Large	76.33%	45.00%	29.29%
B	Medium	Small & Large	77.76%	49.80%	32.45%
C	Large	Small & Medium	75.71%	46.73%	30.71%
D	Medium & Large	Small	78.57%	46.53%	32.86%
E	Small & Large	Medium	79.59%	55.10%	34.29%
F	Small & Medium	Large	79.39%	36.12%	24.69%

Table 3.4: Generalizability across scales tests with the gaze-based and sketch-based feature representations.

As we substantiate in detail in the next two subsections, our algorithm shows excellent generalization across scales. Furthermore, gaze features exhibit clear and definite superiority over sketch-based features in terms of generalization power. In other words, mismatch across task scales has no significant effect on the prediction accuracy for gaze features whereas mismatch results in a significant deterioration of the prediction accuracy for all sketch-based features in consideration (Fig. 3.18).

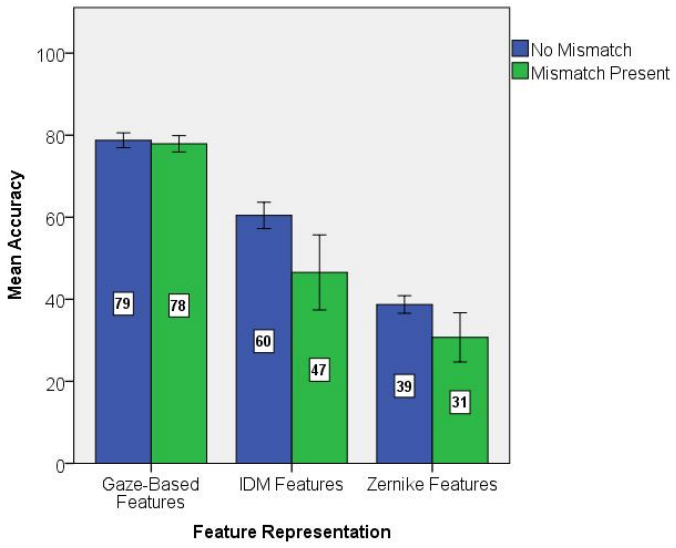


Figure 3.18: Results showing the effects of mismatch presence on prediction accuracy. Our gaze-based feature representation is robust to mismatch across task scales. Error bars indicate 95% confidence interval.

Tests for Assessing Generalizability Across Scales with the Gaze-Based Feature Representation

In order to compare the effect of mismatch on prediction accuracy in match and mismatch conditions, we conducted a one-way between subjects ANOVA. There was no significant effect of mismatch on prediction accuracy at the $p < .05$ level for the two conditions $[F(1, 48) = 0.392, p = 0.534]$.

In order to further compare the effect of mismatch type on prediction accuracy in the six conditions (Table 3.4), we conducted another one-way between subjects ANOVA. There was again no significant effect of mismatch type on prediction accuracy at the $p < .05$ level for the six conditions $[F(5, 24) = 0.405, p = 0.840]$.

Tests for Assessing Generalizability Across Scales with the Sketch-Based Feature Representations

In order to compare the effect of mismatch on prediction accuracy in match and mismatch conditions, we conducted a one-way between subjects ANOVA with IDM Features/Zernike Moments. For both feature representations, there was a significant effect of mismatch on prediction accuracy at the $p < .05$ level for the two conditions. More specifically $[F(1, 48) = 6.100, p = 0.017]$ for IDM Features and $[F(1, 48) = 4.662, p = 0.036]$ for Zernike Moments.

In order to further compare the effect of mismatch type on prediction accuracy in the six conditions (Table 3.4), we conducted another one-way between subjects ANOVA with IDM Features/Zernike Moments. For both feature representations, there was no significant effect of mismatch type on prediction accuracy at the $p < .05$ level for the six conditions. More specifically $[F(5, 24) = 0.285, p = 0.917]$ for IDM Features and $[F(5, 24) = 0.195, p = 0.962]$ for Zernike Moments.

Chapter 4

GAZE-BASED INTELLIGENT USER INTERFACE

For several years, great effort has been devoted to developing gaze-based prediction models that capture human behavior patterns naturally accompanying virtual interaction tasks such as reading an electronic document, or manipulating a virtual object (Fig. 4.1) [3, 6, 11, 16, 19, 74].

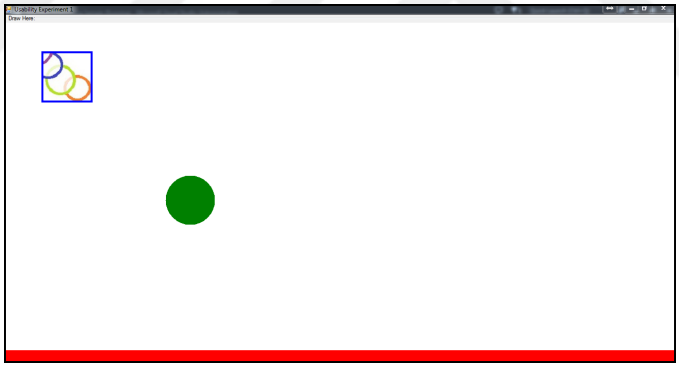
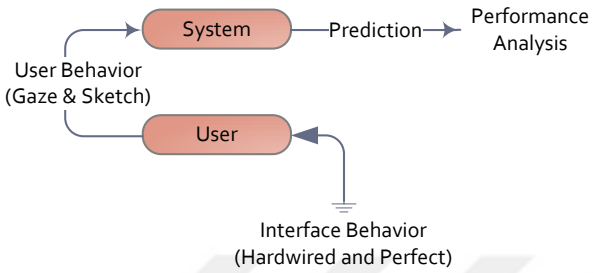


Figure 4.1: Screen capture of one of our predictive user interfaces visualizing an example virtual interaction task. User's task is to drag the blue square (located on the upper-left of the screen) onto the center of the green circle (located on the bottom-right of the screen). We use our gaze-based virtual task prediction model to predict user's task-related intentions and goals in real-time. Furthermore, we assist the user by automatically triggering various user interface adaptations that reflect these predictions.

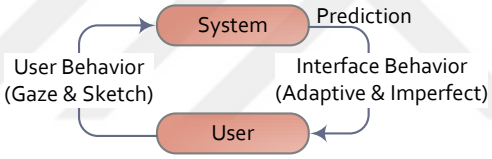
However, existing models are generally evaluated in terms of prediction accuracy, and within offline scenarios that assume perfect knowledge about user's task-related intentions and goals. Such scenarios are called wizard-based test scenarios. In an example offline wizard-based test scenario, the users are asked to either select an object, or to manipulate a previously selected object [3]. Collected data with labels corresponding to user intentions are then used to compute the accuracy of the re-

lated intention prediction model. The output of the prediction model is in no way shown to the users. In other words, in the wizard-based test scenarios, the loop between the user and the prediction system is open, i.e. the user is fed hardwired and perfect visual feedback via the user interface irrespective of predictions made by the prediction system (Fig. 4.2a). Existing studies do not take into account how these models would perform in the absence of wizards. They also do not examine how/if the prediction errors affect the quality of interaction. In this thesis, we eliminate the wizard assumption and close the loop between the user and the prediction system by feeding highly accurate but imperfect predictions (since we do not have prediction systems that can perform with 100% accuracy yet) made by the prediction system to the user via appropriate visualizations of the user interface (Fig. 4.2b). By means of a thorough usability study, we seek answers to the following research questions: (1) How should a user interface adapt its behavior according to real-time predictions made by the underlying prediction system? (2) Will adaptations affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior)? (3) Will prediction errors affect user behavior and inhibit performance of the prediction system? (4) Does users' compatibility with the prediction system have an impact on the design of such interfaces?

We have five main contributions. First, we present the initial line of work that uses real-time feedback generated by a gaze-based probabilistic task prediction model to build an adaptive real-time visualization system. Our system is able to dynamically provide adaptive interventions that are informed by real-time user behavior data. Using a probabilistic model (that is realistically less than 100% accurate) as the basis of a visualization system poses challenges associated with uncertainty. More specifically, the challenge here is to find a novel way of providing feedback about user's task-related intentions and goals throughout a task without being 100% sure of user's intentions. This brings us to our second contribution. We propose two novel adaptive visualization approaches that take into account the presence of uncertainty in prediction model outputs. Our interfaces visualize effects of all possible actions



(a) Open-Loop System



(b) Closed-Loop System

Figure 4.2: Closing the loop between the user and the prediction system.

simultaneously for the duration of an action. When the action is finalized, irrelevant effects disappear and only the effects of the intended action remain visible. We believe that the user's eyes will focus on the effects of the intended action and irrelevant effects will not affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior). Third, we offer a personalization method to suggest which adaptive visualization approach will be more suitable for each user in terms of system performance (measured in terms of prediction accuracy). Personalization boosts system performance and provides users with the more optimal visualization approach (measured in terms of usability and perceived task load). Fourth, by means of a thorough usability study, we provide answers to the questions of whether the proposed visualization approaches or prediction errors affect natural

user behavior and inhibit performance of the underlying prediction systems. This paper also serves to demonstrate that our gaze-based task prediction system detailed in Chapter 3 that was assessed as successful in an offline test scenario, can also be successfully utilized in realistic usage scenarios.

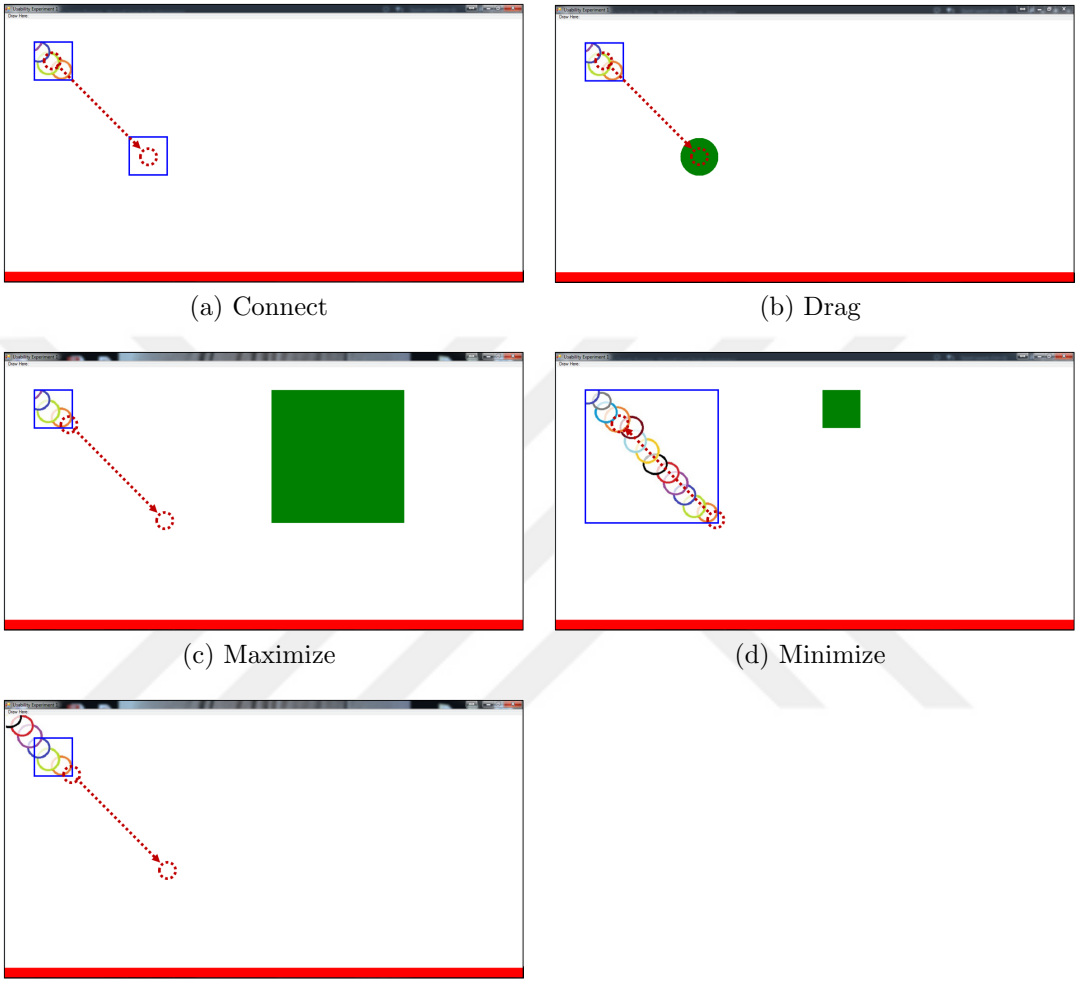
Chapter 4.1 provides details on our usability study, proposed adaptive visualization approaches, and proposed gaze-based predictive user interfaces. Chapter 4.2 describes the evaluation of our predictive user interfaces in terms of performance, usability, and perceived task load.

4.1 Usability Study

Consider the tasks described in Fig. 4.3. We have a gaze-based virtual task prediction system that can accurately distinguish between these tasks. In this thesis, we propose to use online feedback from this system to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. This gives rise to the following research questions: (1) How should a user interface adapt its behavior according to real-time predictions made by the underlying prediction system? (2) Will adaptations affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior)? (3) Will prediction errors affect user behavior and inhibit performance of the prediction system? (4) Does users' compatibility with the prediction system have an impact on the design of such interfaces?

4.1.1 Demographics

We conducted our usability study on 19 participants (17 males, 2 females) recruited from undergraduate and graduate students of our university's engineering faculty on a voluntary basis. Our participants were aged 20 to 26 years old (M = 23.3, SD = 2.0). 10 participants had normal vision, while the remaining 9 had corrected-to-normal vision. 15 participants had dark-colored eyes, while the remaining 4 had fair-colored eyes. On a scale between 1 (none) to 5 (application developer), participants were



(a) Connect (b) Drag

(c) Maximize (d) Minimize

(e) Scroll

Figure 4.3: Pen-based virtual interaction tasks included in our research. Demonstrative examples of how each task can be performed are visualized with dotted visualizations. Starting and ending positions of the exemplary pointer motion is visualized with dotted circles whereas direction of the exemplary pointer motion is visualized with a dotted arrow connecting the starting and ending positions. It is important to note that the dotted visualizations only serve as a reference within this document, and they are not meant to be shown to the user during the usability study.

moderately experienced with tablets (M = 3.7, SD = 0.9), and less so with pen-based tablets (M = 2.4, SD = 1.2) and eye trackers (M = 2.5, SD = 0.8).

4.1.2 Setup

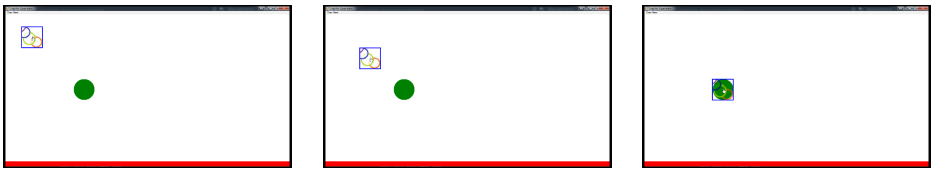
We used a Tobii X120 stand-alone eye tracker and a tablet to collect synchronized gaze and pen data, respectively. Tobii X120 operates with a data rate of 120 Hz, tracking accuracy of 0.5°, and drift of less than 0.3°. The tracker allows free head movement inside a virtual box with dimensions 30×22×30 cm. For displaying our user interfaces accompanied by user's pen position on the tablet, we used a 18.5" Samsung wide screen LED monitor connected to a PC with Intel Core i5-2500 3.30 GHz CPU and 8 GB RAM. Our interfaces were implemented in C++ language using the Visual Studio 2013 IDE.

4.1.3 User Interfaces

To find answers to the before-mentioned research questions, we have designed and implemented 5 different user interfaces that collectively serve as a generalized, context-free, and non-application-specific test bed. The first two are wizard-based interfaces and will be respectively referred to as *wizard UI*, and *after-the-fact wizard UI*. Wizard-based interfaces assume that there exists a "wizard" which knows and informs the underlying prediction system about the user's intentions, thereby allowing the system to provide the user with correct visual feedback at any moment during interaction. The remaining three are realistic predictive interfaces that eliminate the wizard assumption and will be respectively referred to as *after-the-fact predictive UI*, *real-time predictive UI*, and *subtle real-time predictive UI*. Our predictive interfaces demonstrate alternative ways of visualizing real-time predictions, and hence each constitute an answer to the first question. To answer the second and third questions, we compare the predictive interfaces with the wizard-based interfaces with respect to system performance (measured in terms of prediction accuracy), usability, and perceived task load. To answer the fourth question, we search for a correlation between users' compatibility with the prediction system and measured performance on different predictive interfaces.

Wizard UI

Wizard UI can be thought of as the "gold standard" among our interfaces. It is designed to resemble as closely as possible the WIMP-based user interfaces that users are familiar with. Accordingly, in this wizard interface, the underlying prediction system has no command over the interface and prediction results are not visualized by means of any interface adaptations. Expectedly, the user is unaware of predictions errors. In other words, the loop between the user and the prediction system is open, i.e. the user is fed hardwired and perfect visual feedback via the user interface irrespective of predictions made by the prediction system (Fig. 4.4). We use the system performance, usability, and perceived task load of this wizard interface as the upper bound and evaluate our proposed predictive interfaces in comparison with this interface. Underlying prediction systems have been trained with multimodal user data previously collected via a nearly identical user interface (that also does not visualize predictions). Therefore, system performance of this interface is expected to surpass others. Usability and perceived task load of this interface is similarly expected to surpass others since it is deliberately designed to resemble WIMP-based user interfaces that users interact with every day for the past 30 years or so.



(a) Onset of Action (b) During Action (c) End of Action

Figure 4.4: Screen captures of *wizard UI* during a *drag* task. Images serve as illustrations of how our interface looks at the onset, during, and at the end of the user's pen action, respectively. Position of the manipulated object changes in accordance with the user's pen action. Note that the user is fed visual feedback about the current task, and that task only.

After-the-Fact Wizard UI

We have a prediction system that can accurately distinguish between intended user actions (i.e. with approximately 90% success rate for 5 actions). Users can greatly benefit from a user interface that reflects user's task-related intentions and goals in real-time. For this purpose, the loop between the user and the prediction system must be closed, i.e. highly accurate but imperfect predictions made by the prediction system must be fed to the user via appropriate visualizations of the user interface. In line with the feedback principle of design [53], the user interface must provide immediate and appropriate visual feedback about the effects of user's actions from the start to the end of an action. However, the prediction system can say its final word on the user's action only once the action is completed. The challenge here is to find a novel way of providing feedback about user's action-related intentions and goals throughout an action without knowing user's intentions. In other words, the challenge is uncertainty visualization.

After-the-fact wizard UI is our first step towards tackling the uncertainty visualization challenge. We propose a novel user interface approach where effects of all possible actions are visualized simultaneously for the duration of an action (Fig. 4.5). When the action is finalized, irrelevant effects disappear and only the effects of the intended action remain visible (Fig. 4.6). We believe that the user's eyes will focus on the effects of the intended action and irrelevant effects will not affect user behavior and inhibit performance of the prediction system (that assumes natural human behavior). This user interface will serve as a means of testing this argument. Note that this interface is also a wizard interface, i.e. once the action is completed, the intended action information is provided by the wizard instead of the underlying prediction system. Accordingly, this user interface is also free from prediction errors.

After-the-Fact Predictive UI

After-the-fact predictive UI can be regarded as a realistic version of *after-the-fact wizard UI*, where the wizard assumption is eliminated and the intended action in-

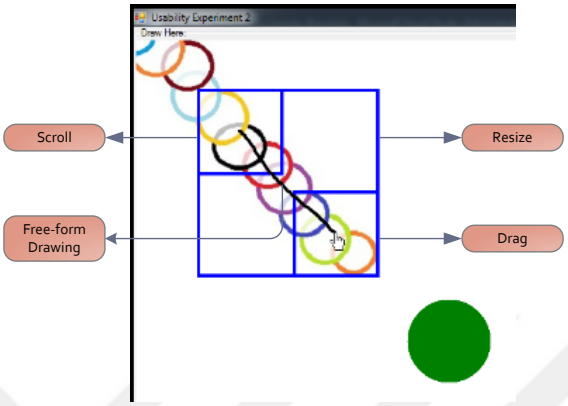


Figure 4.5: We introduce a novel visualization paradigm for gaze-based predictive user interfaces where effects of all possible actions are visualized simultaneously for the duration of an action. This paradigm that we will refer to as *simultaneous visualization* can be utilized for providing visual feedback to users in the presence of uncertainty.

formation is provided by the underlying prediction system instead of the wizard. Accordingly, when the user completes an action, s/he may see effects of an unrelated action due to an erroneous prediction result (Fig. 4.7). This predictive interface that we propose for visualizing prediction results can be employed in an online usage scenario, hence system performance, usability, and perceived task load of this interface is of great interest to our usability study.

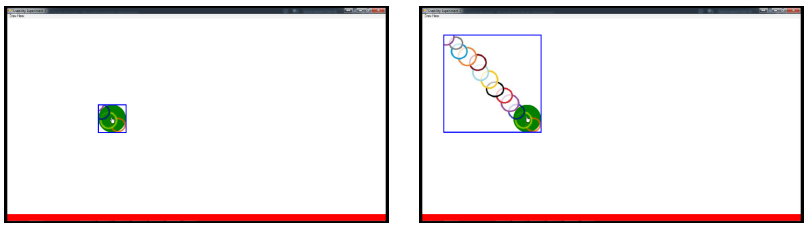
Real-Time Predictive UI

Showing the effects of irrelevant actions for the entire duration of an action can be cumbersome and lead to a heavily cluttered interface as the number of possible actions increases. Albeit not maximally accurate (due to decreasing amounts of behavioral data collected via the related sensors), it is possible to acquire prediction results in real-time from the start to the end of an action. Moreover, since our prediction system is of probabilistic nature, it is also possible to acquire the likelihoods of an action being the intended action in real-time. On that account, we propose another novel

(a) Onset of Action (b) During Action

(c) End of Action (d) After "Prediction"

Figure 4.6: Screen captures of *after-the-fact wizard UI* during a *drag* task. Effects of all possible actions are visualized simultaneously from the onset until the end of the action. When the action is finalized, a prediction is made about the user's intended action. Accordingly, irrelevant effects disappear and only the effects of the intended action (i.e. *drag*) remain visible (Fig. 4.6d). However, there is no prediction really since the intended action information is provided by the wizard.



(a) After a Correct Prediction (b) After an Example Incorrect Prediction

Figure 4.7: Screen captures of *after-the-fact predictive UI* during a *drag* task. Screen captures in Fig. 4.6 also apply to this interface with only one difference. In this case, the intended action information is provided by the underlying prediction system. Hence, when the action is finalized, the user may see effects of an unrelated action due to possible prediction errors. For example, Fig. 4.7b shows what the UI looks like if user's intended action is incorrectly predicted as a *maximize* task instead of a *drag* task.

user interface approach where effects of all possible actions are visualized simultaneously for the duration of an action with dynamically changing levels of transparency.

More specifically, we envision a user interface where increasing levels of transparency indicates decreasing likelihoods of an action being the intended action (Fig. 4.8). This allows us to create a less cluttered and more responsive real-time predictive interface that does not wait until the end of an action to make a prediction.

Every 500 milliseconds, the underlying prediction system feeds the user interface with a list of probability values each denoting the likelihood of an action being the intended action. This, in turn triggers the scene to be redrawn according to the updated likelihood values (Fig. 4.9). We employ the following steps to create a mapping from the likelihood value p to the alpha value α to determine the transparency level of each effect. Likelihood values range from 0 to 1 and alpha values range from 0 to 255 (0 indicating full transparency and 255 indicating full opacity). If we directly map the likelihood values to alpha values, the effect of an action might fully disappear as its likelihood value approaches too close to 0. To make sure that effects of all actions are visible at all times, we increment the likelihood value of each effect with a base likelihood value of 0.25. Note that the initial value of p is set to the base likelihood value for all actions. Then we map the likelihood values to acquire alpha values in the range [64 255] using the following formulas:

$$p = 0.75 * p + 0.25 \tag{4.1}$$

$$\alpha = p * 255 \tag{4.2}$$

Note that a similar methodology applies to the previously described *after-the-fact predictive UI* where the alpha value is fixed at 255, i.e. all effects are fully opaque at all times.

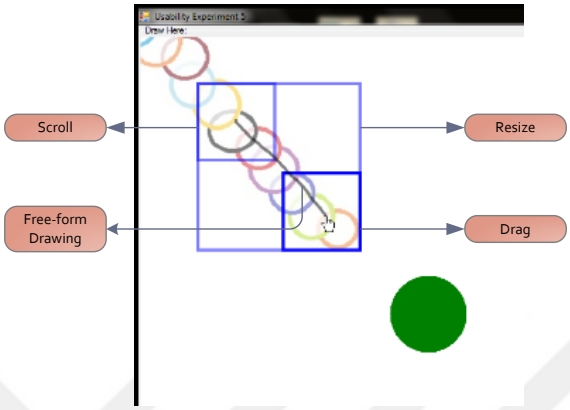


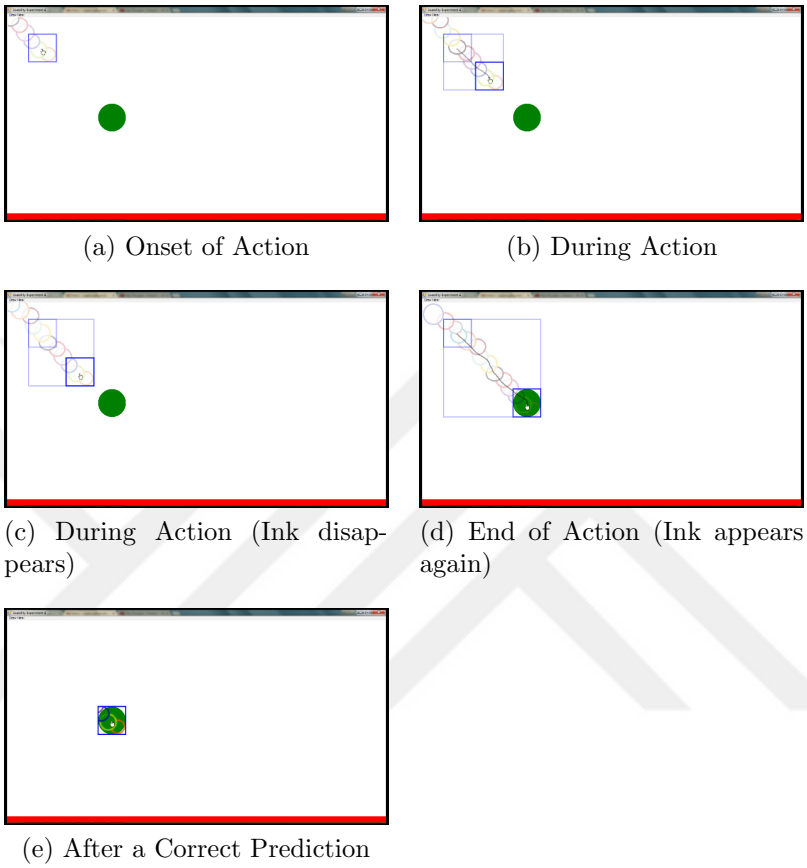
Figure 4.8: We introduce another novel visualization paradigm that we will refer to as *adaptive transparency*. It can similarly be utilized for uncertainty visualization in gaze-based predictive user interfaces. In this paradigm, the user interface dynamically adapts itself according to user's real-time intentions and goals. In this respect, our novel visualization paradigm is similar to as-you-type suggestions (i.e. incremental search or real-time suggestions) used in popular search engines or predictive keyboard applications for mobile devices.

Subtle Real-Time Predictive UI

Subtle real-time predictive UI can be regarded as a more subtle version of *real-time predictive UI*, where the base likelihood value is twice as large, and hence the range of alpha values starts at a higher level. In this case, the likelihood values are mapped in a similar fashion to acquire alpha values in the range [128 255] using the following formulas:

$$p = 0.50 * p + 0.50 \tag{4.3}$$

$$\alpha = p * 255 \tag{4.4}$$



(a) Onset of Action

(b) During Action

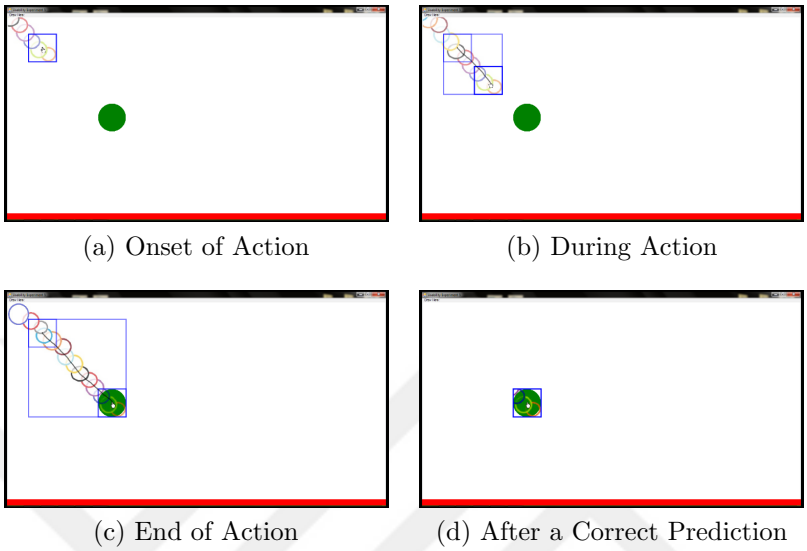
(c) During Action (Ink disappears)

(d) End of Action (Ink appears again)

(e) After a Correct Prediction

Figure 4.9: Screen captures of *real-time predictive UI* during a *drag* task. Effects of all possible actions are visualized simultaneously from the onset until the end of the action. These effects have dynamically changing levels of transparency indicating the likelihood of each action being the intended action at any instant during interaction. It is possible for effects of unlikely actions to disappear as in Fig. 4.9c based on the instantaneous prediction results. Visibility fluctuation may be found plausible by some users and distracting by others, further analysis in Chapter 4.2 will seek an answer to this question among others.

This increase in the base likelihood value results in decreased fluctuation of transparency levels, and hence a more stable interface. Note that similarly, the initial value of p is set to the base likelihood value for all actions (Fig. 4.10).



(a) Onset of Action (b) During Action

(c) End of Action (d) After a Correct Prediction

Figure 4.10: Screen captures of *subtle real-time predictive UI* during a *drag* task. Similarly, effects of all possible actions are visualized simultaneously with dynamically changing levels of transparency. When compared with the previous interface, effects of all actions are more pronounced at all times and it is not possible for effects of unlikely actions to disappear, both due to the increase in the base likelihood value.

4.1.4 Procedure

Each participant was assigned to each of the five user interface conditions, resulting in a repeated measures design. The order of conditions presented to each participant was randomized based on the Latin square method (using a 5x5 Latin square). During each condition, participants were instructed to complete 5 randomized repeats of 5 tasks (Fig. 4.3). The order of tasks presented during each condition was randomized as well. It took each participant about 30 minutes to complete the study. By means of our usability study, we compiled a database of eye gaze, pen, and predicted task label data from 19 participants for 5 randomized repeats of 5 tasks in 5 different user interface conditions. In-between the conditions, participants received 5 practice runs corresponding to each of the 5 tasks in the upcoming user interface condition.

Overall, our usability study consisted of 4 main stages. In the **first stage**, participants were presented with the study guidelines. During this stage, we informed the participants in advance about the various visual effects they might face while

performing the tasks (such as visual feedback corresponding to unrelated tasks, or changes in transparency). More specifically, we asked them to concentrate on the given tasks emphasizing the fact that these effects did not determine or affect their success by any means. In addition to this, we requested the participants to keep their eyes on the display device, use a single stroke to complete each task, and maintain an appropriate distance to the eye tracker (which could be monitored and adjusted via the status bar that stayed green as long as the participant was inside the tracking range). In the **second stage**, participants were asked to complete a standard 9-point calibration procedure posed as an "attention test" in order to conceal any hints of eye tracking. **Third stage** was the main data collection stage. Participants received the tasks one by one. At the beginning of each task, prerecorded non-distracting (in terms of avoiding unsolicited gaze behavior) audio instructions were delivered via headphones. Transcripts of the audio instructions given to the participants for each task are listed as follows[5]:

- Connect: Connect the centers of the two squares
- Drag: Drag the blue square onto the center of the green circle
- Maximize: Increase the size of the blue square to match the size of the green square
- Minimize: Decrease the size of the blue square to match the size of the green square
- Scroll: Pull the chain until the color of the last link is clearly visible

For each task, participants are asked to manipulate the object in a certain way. Desired pen motion starts at the center of the object and follows a diagonal line of 10.5 cm. However, when participants manipulate the task as they see fit, the task is assumed to be complete. We believe this flexibility in task completion criteria is necessary to elicit natural behavior from participants. In order to manipulate the object,

[5]Note that the instructions for the drag, maximize, and minimize tasks contain color information which will not show in a B/W copy of Fig. 4.3. For these tasks, the object to be manipulated (dragged/maximized/minimized) is the one on the left side of each screen.

participants use the pen-based tablet and the display. A hand-shaped visual cursor is rendered on the display to indicate the position of the user's pen on the tablet. If anything went wrong during a task (e.g. the percentage of gaze data flagged *valid* by the eye tracker was less than 80%, or the participant accidentally made redundant/irrelevant pen movements), the current task was repeated. In the **fourth and final stage** of our usability study, a questionnaire was handed over to participants to collect qualitative data about the usability and perceived task load associated with our user interfaces as well as demographic data. For the questionnaires, we gathered our user interfaces into three groups: first group consisted solely of *wizard UI*, second group consisted of the after-the-fact interfaces, and third group consisted of the real-time interfaces. Therefore, users were asked to submit three answers instead of five to each of the questionnaire items. This grouping approach is necessary since users cannot differentiate between different flavors of after-the-fact and real-time interfaces without knowing further details about our usability study, perhaps the most important being the presence of underlying prediction systems. For the questionnaire, we compiled a series of Likert-type questions based on the System Usability Scale (SUS) [8] and the NASA Task Load Index (NASA-TLX) [31] assessment tools. SUS gives a high-level subjective view of usability while NASA-TLX rates perceived workload. Both tools allow the researchers to add scores of individual questions to yield a single score on a scale of 0-100. Since some questions (e.g. "How much physical activity was required?") are irrelevant to our usability study, we have excluded them from our questionnaire. As a result, we included the following list of questions in our study:

SUS Questions to Assess Usability (With Items on a 5-Point Likert Scale)

- I thought the system was easy to use.
- I found the system unnecessarily complex.
- I would imagine that most people would learn to use this system very quickly.
- I thought there was too much inconsistency in this system.

- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

(Note that positively- and negatively-worded questions are alternated so that the participants have to read each statement and make an effort to think whether they agree or disagree with it.)

TLX-NASA Questions to Assess Perceived Performance, Effort, and Frustration (With Items on a 20-Point Likert Scale)

- How successful were you in accomplishing what you were asked to do?
- How hard did you have to work to accomplish your level of performance?
- How insecure, discouraged, irritated, stressed, and annoyed were you?

4.1.5 Underlying Gaze-Based Task Prediction Systems

In the previous subsections, we have repeatedly referred to *underlying prediction systems* that provide intended action information to our user interfaces. These systems are in fact statistical prediction models trained with machine learning algorithms on previously collected user data. In total, we have two major task prediction systems: an after-the-fact prediction system, and a real-time prediction system. The former is integrated into our *after-the-fact predictive UI* whereas the latter is integrated into our *real-time predictive UI* and *subtle real-time predictive UI*. In this subsection, we describe these systems in detail.

After-the-Fact Task Prediction System

Our after-the-fact prediction system builds upon our gaze-based virtual task prediction system detailed in Chapter 3. We modify the existing system slightly to the needs of a responsive real-time user interface. More specifically, we decrease the average time it takes for the existing system to determine the type of a newly completed action from 1.125 seconds to 0.039 seconds.

Our after-the-fact prediction system waits until the ongoing action is completed to provide intended action information. It outputs a single value denoting the predicted action from the list of possible actions. More specifically, it outputs a single value from the set $\{1, 2, 3, 4, 5\}$ since we have five tasks in total. To determine the type of a newly completed action, this system extracts three kinds of features from the collected gaze and pen (sketch) data. These features are: (1) evolution of instantaneous sketch-gaze distance over time, (2) spatial distribution of gaze points collected throughout a task, and (3) IDM visual sketch features [57]. Detailed description of each feature can be found in Chapter 3.

We focus on optimizing the computational time of the first feature since we have previously demonstrated that it is this feature (more specifically the Dynamic Time Warping (DTW) library it utilizes) that causes the performance bottleneck [17]. The first feature models the time-wise evolution of the instantaneous distance between pen tip and gaze direction over time using a time-series signal. Initially, one or multiple characteristic signals are computed per task (Fig. 4.11). When it comes to determining which task a new signal belongs to, similarity of the new signal to each of the characteristic signals is measured. For computing the similarity of two given signals, an open-source MATLAB-based DTW is used [28]. To reduce the time requirement of this similarity computation, we have replaced the MATLAB-based library with another library that is written and compiled in the more efficient C programming language [22]. Numerically, this allows us to process a single action in 0.039 seconds instead of 1.125, an improvement by a factor of approximately 30 times.

Using the optimized version of our feature extraction mechanism, we train our after-the-fact prediction system following the standard three-step machine learning pipeline. The first step involves extracting feature vectors from a set of data samples. To this end, we extract the before-mentioned three kinds of features to obtain three separate feature vectors for each completed action in the database. The first two feature vectors are combined via feature-level fusion and the third feature vector is merged with this combination via classifier-level fusion, both decisions taken based

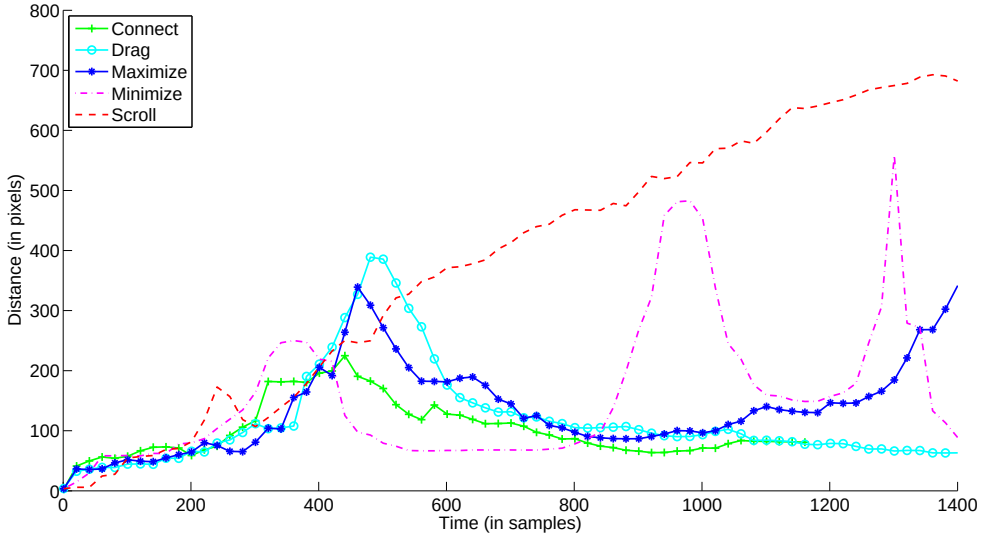


Figure 4.11: Characteristic signals obtained from sketch-gaze distance signals of each task.

on our previous findings on how information fusion technique effects accuracy values in our context [16]. Note that for extracting the feature vectors, we use the same set of data samples detailed in Chapter 3. The second step of the pipeline involves training prediction models using the extracted feature vectors. For this purpose, we train a single Support Vector Machines (SVM) model using the Gaussian radial basis function (RBF) kernel. In this step, we do not partition the input data into disjoint folds for training and testing, and instead use the whole data for training our model since we will use real-time user data during the usability study for testing purposes, which in fact constitutes the third and final step of the pipeline.

Real-Time Task Prediction System

Our real-time prediction system provides on-the-fly intended action information from the start to the end of action. It outputs a list of probability values each denoting the likelihood of an action from the list of possible actions being the intended action. More specifically, it outputs five likelihood values each in the range [0 1] since we have

five tasks in total.

Training of our real-time prediction system is similar to the training of our after-the-fact prediction system except for one major difference. We use our real-time prediction system to create responsive interfaces that dynamically adapt themselves according to user's real-time intentions and goals, and do not wait until the end of an action to make a prediction. This requires a specialized training approach as we have previously proposed in [17]. In line with this approach, we generate 10 different data samples from each individual data sample used in training our after-the-fact prediction system. These 10 samples (that we refer to as *sub-samples*) correspond to the first 10%, 20%, ..., 100% of the original data sample in terms of time elapsed from the start of the corresponding task. The sub-samples created from the set of all samples are then separated into five different groups as follows (note that a typical task lasts about 2 seconds):

- First group consists of sub-samples that last shorter than 500 milliseconds ($0 <= duration <= 500$),
- Second group consists of sub-samples that last shorter than 1000 milliseconds ($500 < duration <= 1000$),
- Third group consists of sub-samples that last shorter than 1500 milliseconds ($1000 < duration <= 1500$),
- Fourth group consists of sub-samples that last shorter than 2000 milliseconds ($1500 < duration <= 2000$), and
- Fifth group consists of sub-samples that last longer than 2000 milliseconds ($duration > 2000$).

After the groups are formed, we train a separate SVM model for each group using the sub-samples comprising each group. Accordingly, the first model captures the characteristics of each task in the first 500 milliseconds while the second model captures the characteristics of each task in the first x milliseconds where x is between 500 and 1000, etc. Our real-time prediction system in fact consists of these five

separate SVM models. Every 500 milliseconds, our real-time prediction system uses the appropriate SVM model to compute and feed the user interface with a list of probability values each denoting the likelihood of an action being the intended action. This, in turn triggers the scene to be redrawn according to the updated likelihood values.

On the other hand, similar to the after-the-fact prediction system, the real-time prediction system uses SVM models trained using the Gaussian radial basis function (RBF) kernel, and uses the whole data for training the models instead of partitioning the input data into disjoint folds for training and testing. Moreover, our real-time prediction system uses the same kinds of features for feature extraction, and combines separate feature vectors using the same information fusion techniques. Computational time is ever more important since our real-time prediction system is specifically trained to enable responsive interfaces. Therefore, for the first kind of feature, the same optimized DTW library is used (Fig. 4.12).

4.2 Evaluation

We have proposed five different user interfaces. The first two are wizard-based interfaces. Particularly, the first interface is the "gold standard" among our interfaces due to its deliberate resemblance to the WIMP-based user interfaces that users are accustomed to. Despite their advantages, wizard-based interfaces are not suited to realistic usage scenarios since they assume perfect knowledge about user's action-related intentions and goals. The reality, however, dictates uncertainty about user's intentions and goals unless we have prediction systems that can perform with 100% accuracy. The remaining three interfaces are predictive interfaces. They have each been designed with the goal of building an adaptive user interface that visualizes user's intentions and goals in the presence of uncertainty. In this section, we evaluate the predictive interfaces relative to the wizard interfaces, taking the performance (measured in terms of prediction accuracy), usability, and perceived task load of the first wizard interface as the upper bound. Hence, we both formally test our underlying prediction systems

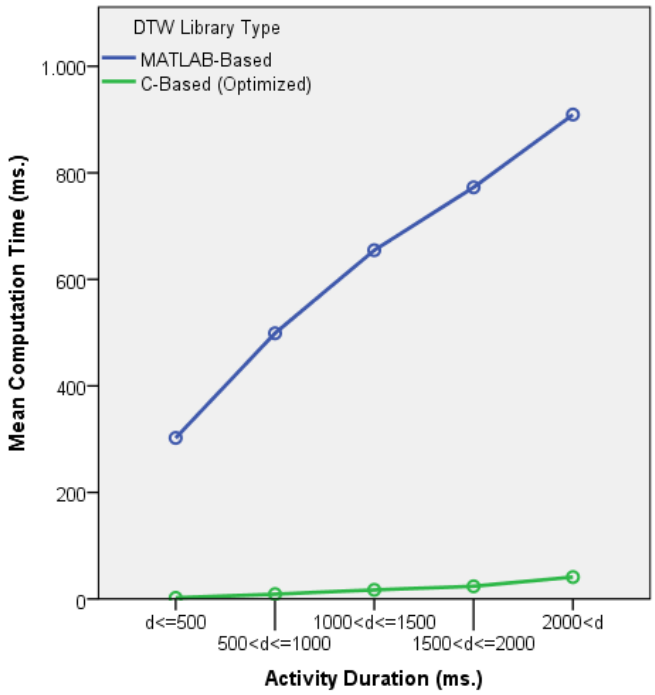


Figure 4.12: Mean computation times obtained with each DTW library as a function of time elapsed from the start of a task. Note that with the MATLAB-based DTW library, it is not even possible to update the user interface every 500 milliseconds according to user's real-time intentions and goals since after a point, it takes more than 500 milliseconds for the prediction system to determine the likelihood values for the ongoing action.

in reasonable scenarios that eliminate the wizard assumption, and propose multiple solutions to the uncertainty visualization challenge faced while designing predictive user interfaces.

We present our evaluation results under four main titles. In Chapter 4.2.1, we compare our interfaces quantitatively and qualitatively without taking subjective differences into consideration, i.e. by inspecting significant differences between mean scores of each user interface averaged over all users. Then in Chapter 4.2.2, we demonstrate that subjective differences are too prominent and significant to be overlooked in the context of our usability study. Therefore in Chapter 4.2.3, we repeat the quantitative and qualitative analysis using a repeated measures design. Taking the subject-based

analysis one step further, we offer a statistical method to predict which predictive user interface will be more suitable for each user in terms of system performance. This personalized approach boosts system performance and provides users with the more optimal interface.

4.2.1 Subject-Independent Results

Quantitative (Accuracy)

We have *intended* and *predicted* task label data collected from 19 participants for 5 randomized repeats of 5 tasks in 5 different user interface conditions. For each user interface, we compute the marginal mean of accuracy by taking the percentage of correctly predicted tasks over all 475 tasks (Fig. 4.13). *Wizard UI* has the highest accuracy among the others. As we have previously mentioned, superior performance of *wizard UI* is expected due to the fact that the underlying prediction systems have been trained with multimodal user data previously collected via a nearly identical user interface (that also does not visualize predictions). More specifically, *wizard UI* and our previously-published user interface both do not involve simultaneous effect visualizations, adaptive changes in transparency, or erroneous predictions. Despite the similarity of these interfaces, accuracy of *wizard UI* is 73% whereas accuracy of our previously-published interface has been reported as 88% [16]. We believe this difference is caused by the fact that *wizard UI* was tested on a different group of participants than the one which provided the multimodal data for training and testing our previously-published interface. This performance degradation can conceivably be avoided by training the underlying prediction systems using only the current user's data or data collected from users who exhibit similar hand-eye coordination behaviors to the current user's.

Following *wizard UI*, *subtle real-time predictive UI* has the second highest accuracy, surpassing even *after-the-fact wizard UI* that is free of prediction errors. This indicates that *subtle real-time predictive UI* is the best candidate for solving the uncertainty visualization challenge while minimizing accuracy degradation.

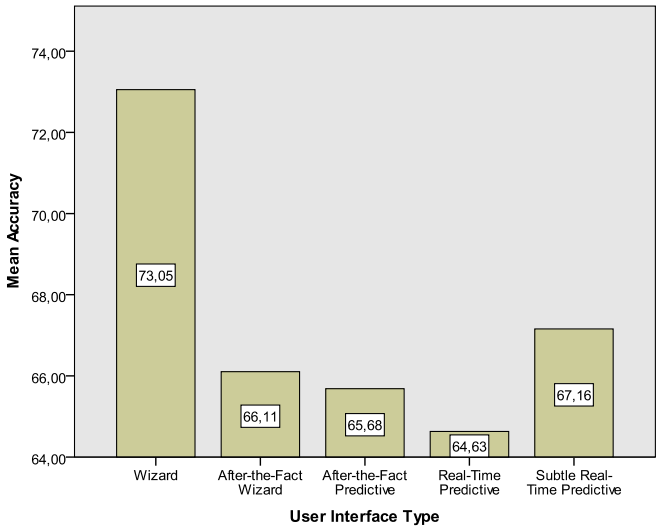
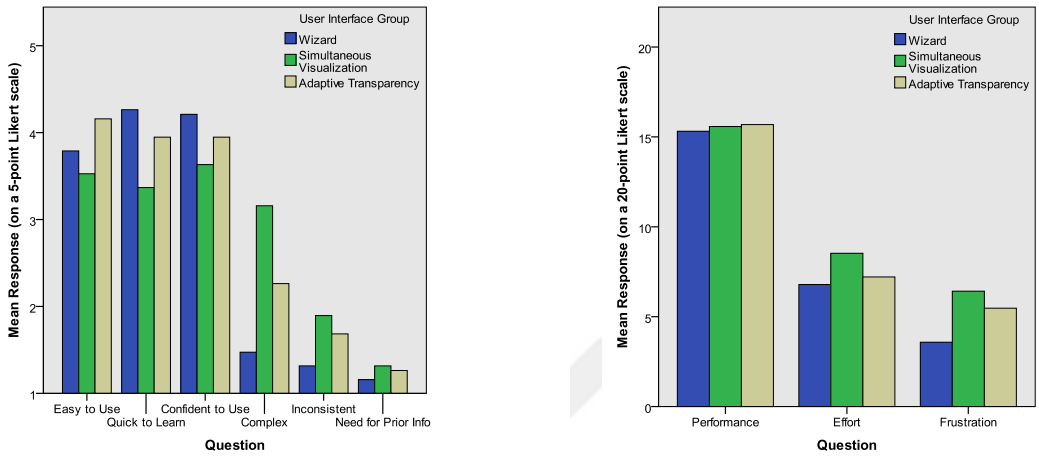


Figure 4.13: Marginal mean accuracy score for each user interface averaged over all users.

Qualitative (Usability and Perceived Task Load)

Overall, usability of the real-time interfaces was rated higher than usability of the after-the-fact interfaces. More specifically, users found the real-time predictive interfaces easier to use, quicker to learn, and they felt more confident using them. In addition, users found the real-time predictive interfaces simpler, more consistent, and they needed less prior information before using them. Likewise, perceived task load of the real-time interfaces was rated lower than the after-the-fact interfaces, i.e. users perceived themselves as more successful in completing the tasks while spending less effort and feeling less frustrated with the real-time predictive interfaces compared to the after-the-fact interfaces.

These results (also summarized in Fig. 4.14) demonstrate that despite the complex mechanisms involved, usability and perceived task load of the real-time predictive interfaces (grouped under adaptive transparency) was rated superior to that of the after-the-fact interfaces (grouped under simultaneous visualization). This indicates that it is beneficial to decrease the clutter and increase the responsiveness of the interfaces by dynamically changing levels of transparency.



(a) Usability (b) Perceived Task Load

Figure 4.14: Marginal mean qualitative results for each user interface measured in terms of usability and perceived task load, and averaged over all users.

4.2.2 A Personalized Approach to Uncertainty Visualization

Performance of a user during interaction with a novel predictive user interface is conceivably linked to the user's *compatibility* with the interface. We use the term *compatibility* to refer to how well the interface collects, reasons about, and visualizes the user's intentions and goals. Highly *compatible* users which receive relatively more accurate feedback about their intentions and goals are more likely to perform better with and have a high opinion about a novel predictive user interface. In addition to our main research questions, we also set aside to find answers to these reasonable claims on personalized differences in *compatibility* with our predictive user interfaces.

Detailed inspection of the accuracy scores reveals high levels of variability among users. Variability is primarily due to subjective differences in *compatibility* with our gaze-based task prediction systems (Fig. 4.15). The majority of users produce information-rich hand-eye coordination behaviors that enable our gaze-based task prediction systems to achieve high accuracy scores irrespective of user interface type. On the other hand, a number of users do not lend themselves well to our gaze-based task prediction systems. Variability is also secondarily due to subjective differences

in user interface inclinations/preferences. For instance, some users are not as affected by prediction errors, others perform better in real-time predictive interfaces compared to after-the-fact predictive interfaces, etc. There is no single common pattern among users summarizing the relationship between user interface type and mean accuracy score. Based on these observations, we take variability among users into consideration when comparing the accuracies of different user interfaces in the following subsections. To this end, we adopt a repeated measures design that provides a way of accounting for variability, thus decreasing non-systematic variance and increasing sensitivity and power of comparisons between different user interfaces. Furthermore, we utilize variability to our advantage by proposing a personalized approach to uncertainty visualization instead of a unified one. This personalized approach fundamentally involves offering each particular user with the user interface that s/he performs better with and prefers more.

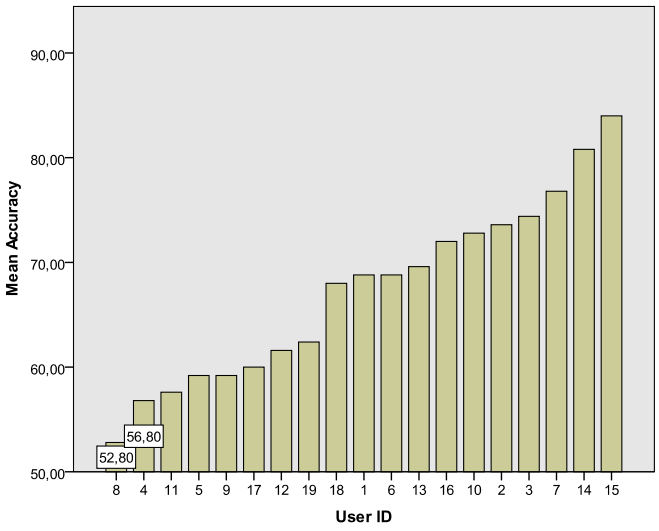


Figure 4.15: Mean accuracy score for each user averaged over all user interfaces. Note that a boxplot analysis of the corresponding data marks the two users with the lowest accuracy scores as mild outliers. However, we have not eliminated their data from future analysis since they are not marked as extreme outliers, and similar users are likely to use our interfaces.

4.2.3 Repeated Measures Design

Quantitative (Accuracy)

Our research primarily seeks answers to the questions of whether user interface adaptations or prediction errors affect user behavior and inhibit performance of the underlying prediction systems (that assume natural human behavior). To find answers to these questions, we conducted a repeated measures ANOVA that compares the effect of user interface type on mean accuracy scores. Mauchly's Test of Sphericity indicated that the assumption of sphericity had not been violated ($\chi^2(9) = 11.918, p = 0.220$), and therefore no corrections were used. There was a significant effect of user interface type on mean accuracy scores, ($F(4, 72) = 3.287, p = 0.016$). Post-hoc tests using the Bonferroni correction revealed that user interface adaptations elicited a slight degradation in accuracy scores for *after-the-fact predictive UI* (65.68 ± 1.99) and *subtle real-time predictive UI* (67.16 ± 3.05) conditions compared to *wizard UI* condition (73.05 ± 2.44). However, both reductions were not statistically significant ($p = 0.15$ and $p = 0.43$, respectively), indicating the suitability of these two predictive interfaces for solving the uncertainty challenge. The reduction was minimal in *subtle real-time predictive UI* condition, further emphasizing the superiority of this user interface. On the other hand, *real-time predictive UI* condition (64.63 ± 2.58) elicited a significant degradation ($p = 0.043$) in accuracy scores compared to *wizard UI* condition, ruling out the candidacy of this interface for solving the uncertainty challenge. Furthermore, there was no significant effect of absence/presence of prediction errors on accuracy scores ($p = 1.00$) across *after-the-fact wizard UI* (66.11 ± 2.73) and *after-the-fact predictive UI* conditions (two conditions that differ only in the absence/presence of an underlying prediction system, and hence of prediction errors). On the basis of these findings, we can conclude that *after-the-fact predictive UI* and *subtle real-time predictive UI* can be used for uncertainty visualization in gaze-based predictive interfaces without significantly affecting user behavior and inhibiting performance of the underlying prediction systems.

Qualitative (Usability and Perceived Task Load)

We have demonstrated that when subjective differences are not taken into consideration, usability and perceived task load of the real-time interfaces are rated superior to usability and perceived task load of the after-the-fact interfaces. In this subsection, we show that repeating the qualitative analysis using a repeated measures design, and hence taking subjective differences into consideration leads us to the same conclusion. To make a concise statement, instead of analyzing responses to individual questions on usability, we compute a single score summarizing all aspects of usability by subtracting the sum of responses to negatively-worded questions from the sum of responses to positively-worded questions[6].

We conducted a repeated measures ANOVA to compare the effect of visualization paradigm on usability. Mauchly's Test of Sphericity indicated that the assumption of sphericity had not been violated ($\chi^2(2) = 2.830, p = 0.243$), and therefore no corrections were used. There was a significant effect of visualization paradigm on usability, ($F(2, 36) = 6.545, p = 0.004$). Post-hoc tests using the Bonferroni correction revealed that usability of *simultaneous visualization* paradigm condition (4.16 ± 4.10) is statistically lower than usability of both "gold standard" (8.32 ± 2.81) and *adaptive transparency* paradigm (6.84 ± 3.69) conditions ($p = 0.016$ and $p = 0.027$, respectively). On the other hand, no significant difference was found between usability of "gold standard" and *adaptive transparency* paradigm conditions ($p = 0.746$). We also conducted a repeated measures ANOVA to compare the effect of visualization paradigm on perceived task performance, however no significant effects were found.

Correlation Analysis and Detection of User Groups Based on Quantitative Evidence

Following the quantitative and qualitative comparative analysis of our user interfaces in a repeated measures design, we created a mapping based on correlation analysis

[6]Positively-worded questions are concerned with ease of use, learnability, and confidence whereas negatively-worded questions are concerned with complexity, inconsistency, and need for prior information. Note that the tools we use for usability and perceived task load assessment [8, 31] allow the researchers to add scores of individual questions to yield a single score.

to predict a user's *compatibility* with our gaze-based task prediction systems based on his/her performance in *wizard UI*. *Compatible* users are assigned to *subtle real-time predictive UI* whereas *incompatible* users are assigned to *after-the-fact predictive UI*. This personalized mapping and subsequent user interface assignment approach enables us to offer each particular user with the user interface that s/he performs better with and prefers more. In this manner, we achieve a mean accuracy score that surpasses the individual mean accuracy scores of both user interface types.

We ran a Pearson product-moment correlation to determine the relationship between a user's mean accuracy score in *wizard UI* and difference between his/her mean accuracy scores in *subtle real-time predictive UI* and *after-the-fact predictive UI*. There was a statistically significant positive correlation ($r = 0.485, n = 19, p = 0.035$). The corresponding linear regression equation (Fig. 4.16) was estimated as follows:

$$\text{Difference} = -32.373 + 0.463 \times \text{Accuracy in Wizard UI} \tag{4.5}$$

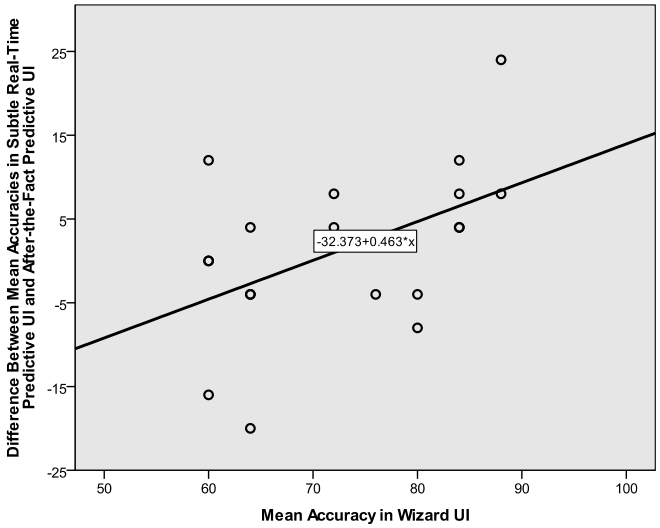
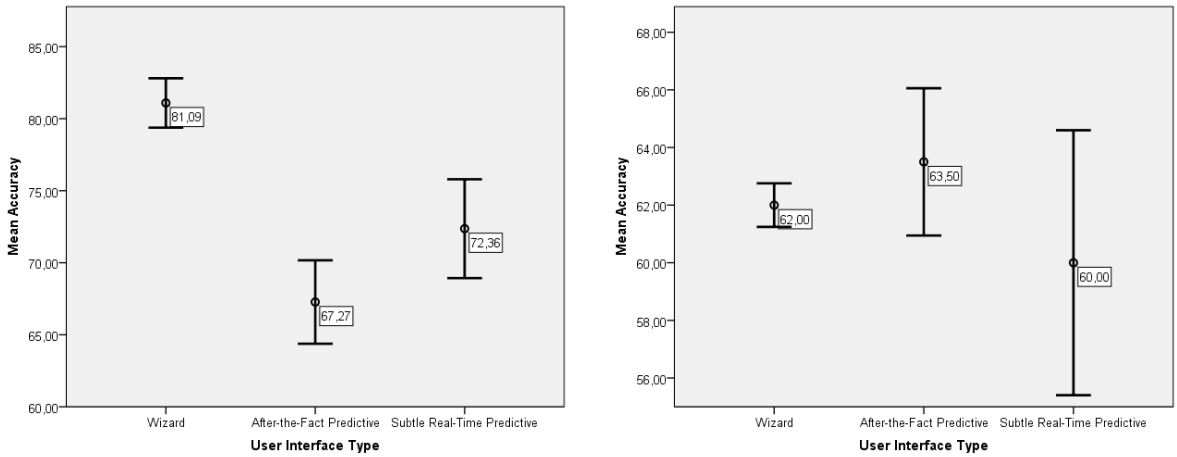


Figure 4.16: Users with high accuracy values in *wizard UI* also have favorable accuracy values in *subtle real-time predictive UI*.

Using this equation and a given user's mean accuracy value in *wizard UI*, we can predict whether the user will perform better in *subtle real-time predictive UI* or *after-the-fact predictive UI*. Since the correlation is positive, users with high accuracy values in *wizard UI* also have favorable accuracy values in *subtle real-time predictive UI*. We refer to users with high accuracy values in *wizard UI* (Difference ≥ 0) *compatible* users and offer them *subtle real-time predictive UI*. On the other hand, we refer to users with relatively lower accuracy values in *wizard UI* (Difference < 0) *incompatible* users and offer them *after-the-fact predictive UI*. This personalized approach yields mean accuracy scores of 72.36% and 63.5% for *compatible* and *incompatible* users, respectively (Fig. 4.17). Averaged over all users, mean accuracy score raises up to 68.63%, surpassing the individual mean accuracy scores of all our predictive user interfaces. Note that the reported mean accuracy scores correspond to the leave-one-out cross-validation accuracy scores since in linear regression we can compute the leave-one-out cross-validation accuracy score for any leave-one-out data set using only a single fit, obtained from all the data [34].



(a) *Compatible* Users (b) *Incompatible* Users

Figure 4.17: Personalization boosts system performance. Note that among our participants, 11 were predicted as *compatible* users and the remaining 8 were predicted as *incompatible* users. Error bars indicate ±1 standard error.

Qualitative Reasoning and Statistical Analysis Behind User Groups

We have created an intelligent system that can predict which user interface a particular user will perform better with based on his/her *compatibility* with our *wizard UI*. More specifically, we offer *subtle real-time predictive UI* to *compatible* users and *after-the-fact predictive UI* to *incompatible users*. In this subsection, we show that in addition to boosting system performance, personalization provides users with the more optimal visualization approach (measured in terms of usability and perceived task load).

Overall, *compatible* users did not prefer the after-the-fact interfaces as much as *incompatible* users. They found these interfaces less easy to use (3.18 vs. 4.00), more complex (3.45 vs. 2.75), and they felt less confident using them (3.36 vs. 4.00). Moreover, they perceived themselves as less successful in completing the tasks (15.00 vs. 16.38) while spending more effort (9.82 vs. 6.75) and feeling more frustrated with these interfaces (7.27 vs. 5.25). We further ran a Pearson product-moment correlation to determine the relationship between a user's rating of usability[7] for the real-time predictive interfaces only and difference between his/her mean accuracy scores in *subtle real-time predictive UI* and *after-the-fact predictive UI*. Note that the latter factor determines the user group of a particular user. There was a statistically significant positive correlation ($r = 0.576, n = 19, p = 0.01$). This further emphasizes the inclination of *compatible* users towards the real-time interfaces.

[7]To make a concise statement, instead of analyzing responses to individual questions on usability, we compute a single score summarizing all aspects of usability by subtracting the sum of responses to negatively-worded questions from the sum of responses to positively-worded questions.

Chapter 5

GAZE-BASED BIOMETRIC AUTHENTICATION SYSTEM

Biometric authentication is the task of determining whether the person is indeed who s/he claims to be. Traditional studies on biometric authentication use one (or a combination) of the following approaches to address this task: (1) proof by possession – focus on what the person owns (e.g. a key, a secure token), (2) proof by knowledge – focus on what the person knows (e.g. a password combination), and (3) proof by biometrics – focus on what is physiologically unique about the person (e.g. iris, fingerprint).

On the other hand, recent studies on biometric authentication focus on using behavioral characteristics such as gait, typing rhythm, and speech dynamics. These studies (that collectively constitute the emerging field of *behaviometrics* [52]) measure and quantify unique human behavioral patterns to verify the identity of a person. Unique behavioral patterns are a result of individual differences in acquired behaviors, style, preferences, knowledge, motor-skills, or strategy used by people while accomplishing different everyday tasks [23]. The fundamental advantage of the studies on behavioral biometrics over the traditional ones stems from the fact that behavioral patterns are inherently very difficult, if not impossible, to forge. For instance, a key can always get stolen, a password combination can always be hacked, and an iris image can always be replicated [20, 39, 66]. However, imitating a person's walking gait involves imitating the precise patterns of how the head, neck, legs, hips, knees, feet etc. move with respect to each other. Therefore, authentication systems based on behavioral characteristics are less susceptible to identity theft when compared with traditional authentication systems.

Eye gaze has fairly recently captured the attention of researchers studying behavioral biometrics. Eye gaze behavior depends largely on the brain activity and extraocular muscle properties of an individual [33]. In other words, eye gaze signal includes both behavioral and physiological human attributes – this possibly very complex mixture makes eye gaze a great candidate for use as an inimitable biometric authentication trait. Two of the most prominent advantages of gaze-based biometrics systems are (1) higher resistance to identity theft due to the inherent difficulty of forging complex gaze patterns, and (2) ability to verify authentication in an implicit, covert, non-intrusive, and contactless manner [39, 43, 65]. In this chapter, we present a gaze-based authentication system that aims to capture these advantages. We have four main contributions:

Our first contribution is a robust gaze-based biometric authentication system. Our approach consists of extracting and creating statistical models for gaze behavior patterns that naturally accompany daily interaction with pointer-based systems including, but not limited to, increasingly prominent pen-based mobiles devices. It is possible to employ our authentication system on its own or as a complementary soft-biometrics system to improve accuracy and counterfeit-resistance.

Our second contribution is a novel set of gaze-based biometric authentication tasks. Users are accustomed to performing simple tasks (e.g. swipe, draw a pattern) to unlock their pen-based mobile devices. We propose to use the following set of tasks for gaze-based authentication: *drag*, *connect*, *maximize*, *minimize*, and *scroll*. As also mentioned by Darwish and Pasquier [20], these kind of tasks are similar to what users are familiar with performing on their pen-based mobile devices. On the contrary, state-of-the-art gaze-based authentication systems revolve around the same unfamiliar task, i.e. viewing a still image of a face for a predefined amount of time [12, 65]. Moreover, our tasks are short (around 2 seconds each) compared to existing tasks that take 4 seconds [65] or 10 seconds [12]. Ease of use and authentication speed are especially important for mobile devices where authentication frequency is high (around 150 times a day [49]), and activities that follow authentication are likely

to be urgent (e.g. calling 911, replying to a text message).

Our third contribution is the first multimodal feature representation for gaze-based biometrics research. All existing feature representations for gaze-based biometric authentication are solely gaze-based. No existing work uses a multimodal feature representation for this purpose. Our feature representation fuses the spatio-temporal information collected via gaze and pointer (or more specifically, pen) modalities in order to verify a user's authenticity. We propose to use three kinds of features, all based on human vision, and behavioral studies. The first kind is of multimodal nature and attempts to capture the dynamic aspects of human hand-eye coordination behavior. The second kind is of unimodal nature and attempts to quantify how the eye gaze data is structured in terms of saccades and fixations (i.e. two main modes of voluntary gaze-shifting mechanism). The third kind is, again, of unimodal nature and attempts to summarize the image-based properties of the pointer data.

Our fourth contribution is the first multimodal database for gaze-based biometrics research. We present a multimodal dataset that consists of gaze and pen input collected from participants completing the before-mentioned authentication tasks using a pen-based interface. This carefully compiled database is the first of its kind, and we believe it will serve as a reference database for future research on gaze-based behavioral biometrics.

The remainder of the chapter is organized into four sections. Chapter 5.1 details our methodology, paying particular attention to our gaze-based biometric authentication tasks, multimodal feature representation, and multimodal database. Evaluation of our gaze-based biometric authentication system is discussed in Chapter 5.2.

5.1 Methodology

We propose a biometric authentication system for pointer-based systems including, but not limited to, increasingly prominent pen-based mobile devices. To unlock a mobile device equipped with our biometric authentication system, all the user needs to do is manipulate a virtual object presented on the device display. The user can

select among a range of familiar manipulation tasks, namely *drag*, *connect*, *maximize*, *minimize*, and *scroll*. These simple tasks take around 2 seconds each, and do not require any prior education or training. More importantly, we have discovered that each user has a characteristic way of performing these tasks. Features that express these characteristics are hidden in the user's accompanying hand-eye coordination, gaze, and pointer behaviors. For this reason, as the user performs any selected task, we collect his/her eye gaze and pointer movement data using an eye gaze tracker and a pointer-based input device (e.g. a pen, stylus, finger, mouse, joystick etc.), respectively. Then, we extract meaningful and distinguishing features from this multimodal data to summarize the user's characteristic way of performing the selected task. Finally, we authenticate the user through three layers of security: (1) user *must* have performed the manipulation task correctly (e.g. by drawing the correct pattern), (2) user's hand-eye coordination and gaze behaviors while performing this task *should* confirm with his/her hand-eye coordination and gaze behavior model in the database, and (3) user's pointer behavior while performing this task *should* confirm with his/her pointer behavior model in the database. Each user who wants authorized access to the related mobile device must initially enroll himself/herself by providing sufficient eye gaze and pointer data for training the models. Our authentication system is closed-set, i.e. we can query whether a person is indeed who s/he claims to be only if this person has previously completed the enrollment process. Please refer to Fig. 5.1 for a depiction of our overall approach to gaze-based biometric authentication.

In the following sub-sections, we detail (1) manipulation tasks that we use for gaze-based biometric authentication, (2) multimodal database that we build from the collected gaze and pointer movement data while users perform these manipulation tasks, (3) features that we extract from the multimodal data to summarize users' characteristic ways of performing these manipulation tasks, and (4) user-specific hand-eye coordination and gaze behavior, and pointer behavior models that constitute our biometric authentication system. Note that in order to build our gaze-based biometric authentication system, we adopt some machinery from our previous study on gaze-

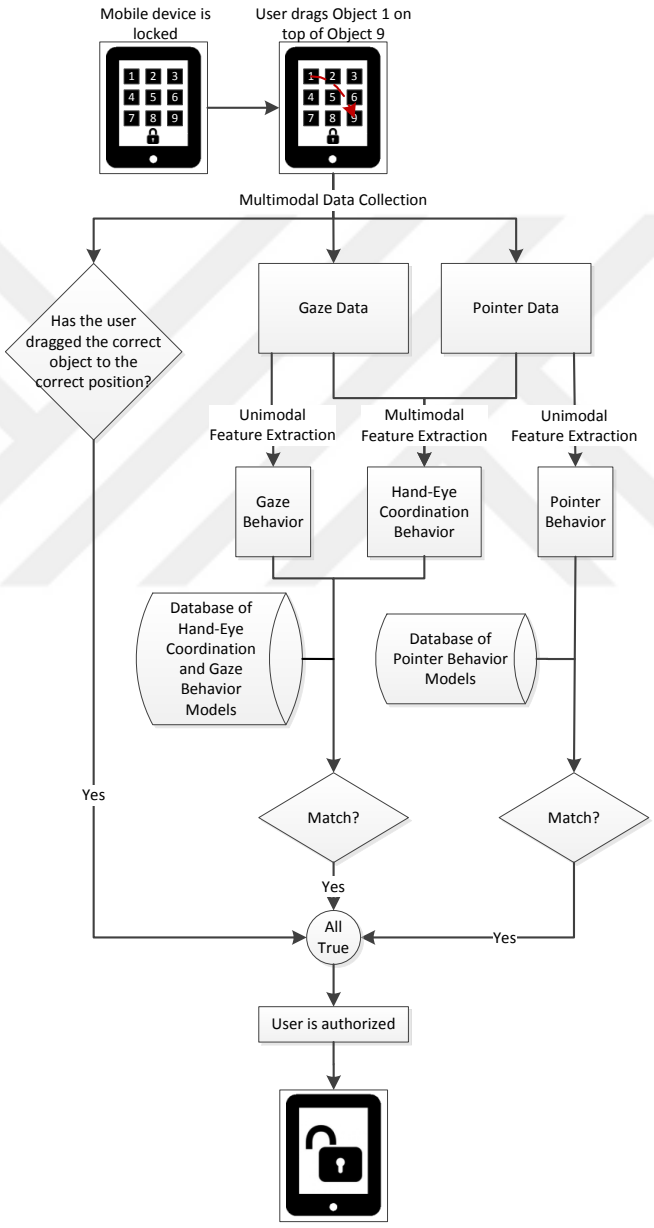


Figure 5.1: Flow diagram visualizing our overall approach to gaze-based biometric authentication. Note that in this diagram, *drag* task is chosen among a range of available tasks for demonstration purposes only.

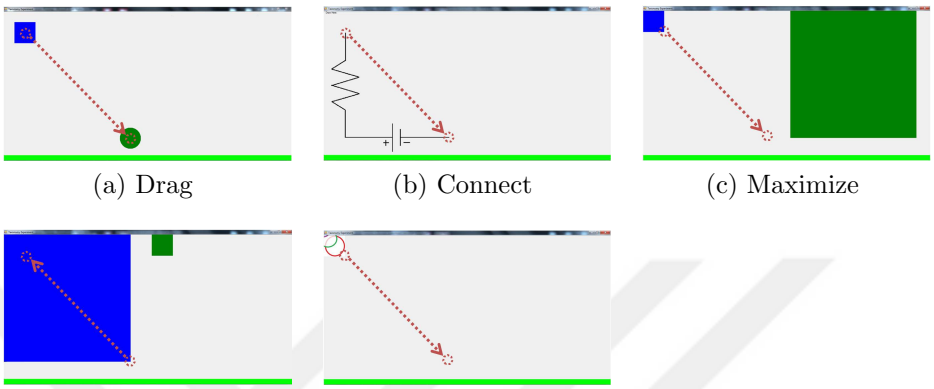
based virtual task prediction [16].

5.1.1 Gaze-Based Biometric Authentication Tasks

Biometric authentication tasks constitute the interface between the user and the authentication system of the related mobile device. Via these tasks, users introduce themselves and gain authorized access to their mobile devices. Again via these tasks, authentication systems elicit and collect meaningful and distinguishing behavioral data from users for user verification. We want tasks that are both familiar to users, easy/quick to perform, and able to consistently elicit characteristic behaviors from users. To this end, we propose a novel set of biometric authentication tasks. For familiarity, ease of use, and authentication speed, our tasks are designed in light of commercially proven methods for unlocking mobile devices (e.g. *slide to unlock* and *draw a pattern to unlock*). For permanence, our tasks are designed as "active tasks" that exploit information emerging from the brain's decision center. More specifically, our tasks necessitate the brain to produce an "automatic schema" [39], and each human being has a slightly different schema marked by his/her customs and habits.

Our first task is the *drag* task (Fig. 5.2a). To accomplish this task, users are asked to "drag the blue square onto the center of the green circle". It resembles the commercially popular *slide to unlock* task during which users are expected to move the unlock image to a predefined location along a predefined path [15]. However, the popular task does not offer any means of security and is merely an intuitive method to activate the mobile device. Moreover, in our task, we do not restrict the users to a predefined path, and instead let them decide freely. We envision more enhanced versions of this task that allow the users to define the initial positions of the blue square and the green circle for improved freedom, and/or present the users with multiple objects without revealing the identities of the source and destination objects for improved security (Fig. 5.3).

Our second task is the *connect* task (Fig. 5.2b). To accomplish this task, users are asked to "connect the battery and the resistor with a wire". It resembles the



(a) Drag (b) Connect (c) Maximize

(d) Minimize (e) Scroll

Figure 5.2: Gaze-based biometric authentication tasks included in our research. Demonstrative examples of how each task can be performed are visualized with dotted visualizations. Starting and ending positions of the exemplary pointer motion is visualized with dotted circles whereas direction of the exemplary pointer motion is visualized with a dotted arrow connecting the starting and ending positions. It is important to note that the dotted visualizations only serve as a reference within this document, and they are not meant to be shown to the user during authentication.

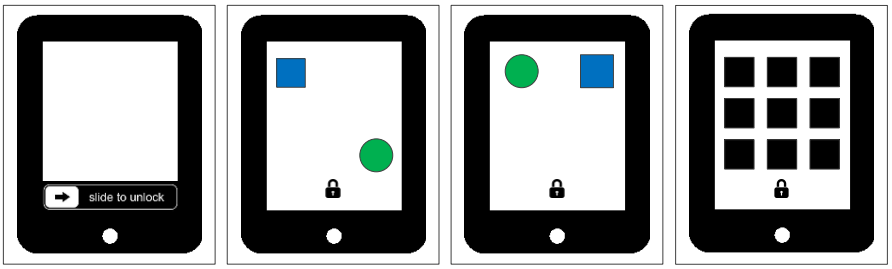


Figure 5.3: Varieties of the *drag* task. From left to right: The commercially popular *slide to unlock* task; our *drag* task with visible source/destination object pair; enhanced version of our *drag* task with freely positioned source/destination object pair; another enhanced version of our *drag* task with multiple objects and hidden source/destination object pair.

commercially popular *draw a pattern to unlock* task during which users are expected to draw a pattern on a grid of 9 dots arranged in a 3 x 3 set. In our task, however, starting and ending positions of the pattern are predefined as the upper left corner and bottom right corner of the grid, respectively. Similarly, we envision more enhanced versions of this task that eliminate this restriction, and/or allow the users to define more complex paths with via points.

Our third and fourth tasks are the *maximize* and *minimize* tasks, respectively (Fig. 5.2c and Fig. 5.2d). To accomplish these tasks, users are asked to "increase/decrease the size of the blue square to match the size of the green square". Similarly, we envision more enhanced versions of these novel resizing tasks where the green square used for reference is not visible and the user resizes the blue square to a preset size self-defined by the user. This enhancement will perceivably allow for a more free and secure authentication system.

Our fifth task is the *scroll* task (Fig. 5.2e). To accomplish this task, users are asked to "pull the chain until the color of the last link is clearly visible". Similarly, we envision a more enhanced version of this novel task where the user determines until which link one needs to pull the chain to unlock the related mobile device.

Mobile devices equipped with our biometric authentication system are planned to operate as follows: Authorized device user picks a task of personal preference among our proposed set of tasks. The user modifies the task with respect to readily available enhancements, e.g. by choosing the destination link of the chain to be pulled in the *scroll* task. The user intuitively, almost instinctively, develops a characteristic schema for performing this task marked by his/her hand-eye coordination, gaze, and pointer behaviors. This characteristic schema is saved to device's database during enrollment process to be later used for deciding whether a person trying to access the mobile device is indeed this same person. On the other hand, an intruder will have a hard time understanding that the object on the screen is the key to unlocking the mobile device, let alone correctly manipulating the object and imitating the authorized user's precise patterns of hand-eye coordination, gaze, and pointer behaviors. To further

confuse the intruder, numbers can be placed on the objects presented on the screen making the authentication system look like a number pad for traditionally entering a 4-digit pin code.

5.1.2 Multimodal Database

We utilize the same multimodal database that we have previously compiled for training and testing probabilistic prediction models that can successfully discriminate between our authentication tasks [16]. In this chapter, we use this database to discriminate between different users instead of between different tasks. First of its kind, we believe this carefully compiled multimodal database will serve as a reference database for future research on gaze-based behavioral biometrics.

With a Tobii X120 stand-alone eye tracker for gaze data and a pen-enabled tablet for pointer data, we have collected multimodal data from each user while they repeatedly perform our authentication tasks. In order to accommodate for mobile devices of different screen sizes, we have created three variations of each task differing from each other only in length of the required pointer motion. The length of the required pointer motion is 21 cm for the *large* scale whereas it is 10.5 cm for the *medium* and 5.25 cm for the *small* scales. In total, we have collected gaze and pointer data from 10 users (6 males, 4 females) over 10 randomized repeats of 5 tasks across 3 scales. This makes up for 1500 *task instances*, where a task instance refers to a single run of a certain task at a certain scale.

5.1.3 Multimodal Feature Representation

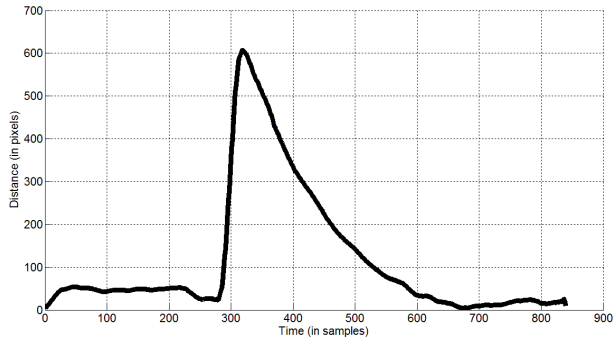
For verifying a user's authenticity, we utilize the same multimodal feature representation that we have previously used to discriminate between our authentication tasks [16]. In this chapter, we use this feature representation to discriminate between different users instead of between different tasks. Our feature representation fuses the spatio-temporal information collected via gaze and pointer (or more specifically, pen) modalities in order to verify a user's authenticity. There is no feature representation in

the literature that fuses information collected via different modalities for gaze-based biometric authentication.

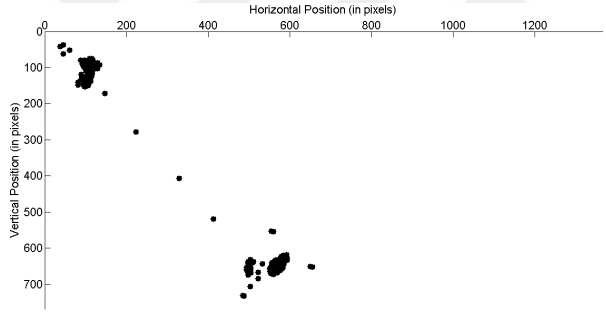
During an authentication task, the following data is available on an eye tracker-enabled, pointer-based mobile device: (1) gaze data, i.e. how the eye gaze points are spatially located on the screen at any point during the task, and (2) pointer data, i.e. what the precise path of the pointing device (e.g. a pen, stylus, finger, mouse, joystick etc.) visually looks like at any point during the task. We propose to extract information-rich features from this time-series data to aid us with biometric authentication. To this end, we use three kinds of features, all based on human vision, and behavioral studies.

The first kind is of multimodal nature and attempts to capture the dynamic aspects of human hand-eye coordination behavior (Fig. 5.4a). Hand-eye coordination behavior inherent in virtual manipulation tasks changes over the course of a task instance as a function of changes in user sub-tasks. The multiple steps of each task can be thought of as consecutive sub-tasks, and each sub-task entails a different type of hand-eye coordination behavior. More importantly, we believe each user adopts a different strategy in terms of how and in which order these sub-tasks are accomplished. Our first feature is based on these observations, and hence attempts to capture the user-dependent dynamic aspects of human hand-eye coordination behavior through the evolution of the distance between instantaneous gaze and pointer positions calculated over a task instance.

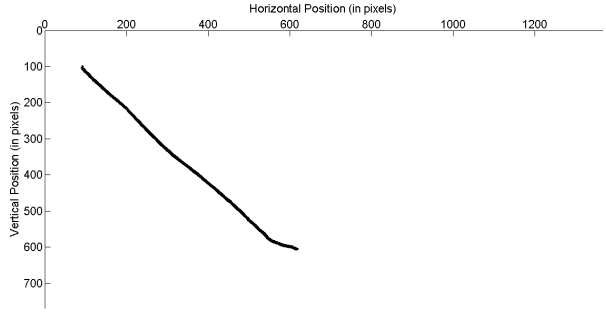
The second kind is of unimodal nature and attempts to quantify how the eye gaze data is structured in terms of saccades and fixations (Fig. 5.4b). Humans employ two different modes of voluntary gaze shifting mechanism to orient the visual axis. These modes are referred to as saccadic and smooth pursuit eye movements. It is widely accepted that "saccades are primarily directed toward stationary targets whereas smooth pursuit is elicited to track moving targets" [21]. Typical virtual manipulation tasks contain both stationary and moving targets. A user's attention can be dominantly directed towards targets of either type depending on the intended



(a) Hand-Eye Coordination Behavior



(b) Gaze Behavior



(c) Pointer Behavior

Figure 5.4: From top to bottom: Plot visualizing how the hand-eye coordination of the user (quantified by the distance between the tip of the pointer and position of the eye gaze) changes with respect to time throughout a task; plot visualizing how the eye gaze points are spatially located on the screen at the end of a task; plot visualizing the precise path of the pointing device at the end of a task. Note that all plots are created based on data extracted from a random *drag* task instance for demonstration purposes only.

task. More importantly, we believe each user adopts a different strategy in terms of allocating attention to which targets and for how long. Our second feature is based on these observations, and hence attempts to quantify how the data is structured in terms of saccades and fixations.

The third kind is, again, of unimodal nature and attempts to summarize the image-based properties of the pointer data (Fig. 5.4c). Even though pointer trajectories for our tasks do not have easily distinguishable characteristic visual appearances when compared with pen trajectories of signatures, sketched symbols, or stylized gestures, it is still conceivable that visual variations in pointer data may aid identity verification. Our third feature is based on these observations, and hence attempts to summarize stroke properties like orientation and endpoint locations using IDM visual sketch features [57]. IDM feature representation has been shown to work well for hand-drawn sketch data [79], and is invariant to rotation and local deformations, making it more tolerant intra-personal visual variations.

Commercially popular state-of-the-art authentication mechanisms ignore these three kinds of biometrically information-rich data that naturally accompany each authentication task. Instead, these systems validate the users only by checking whether the user has correctly completed the authentication task. Authentication tasks, such as *draw a pattern to unlock* can easily be hacked by looking over the shoulder of a user drawing the pattern. We offer to use hand-eye coordination, gaze, and pointer behavior patterns of users during authentication tasks in order to improve accuracy and counterfeit-resistance. Improved security is a natural result of the facts that (1) we now have four, instead of one, way of validating a user, and (2) it is very difficult, if not impossible to hack, steal, or imitate precise patterns of hand-eye coordination, gaze, and pointer behaviors.

5.1.4 Gaze-Based Biometric Authentication System

Our gaze-based biometric authentication system fundamentally consists of a database of user models, i.e. binary classifiers each trained to decide whether a person trying

to access a mobile device is indeed who s/he claims to be. Initially and only once for each user, we create three binary classifiers. If we reiterate the big picture that we have previously presented in Fig. 5.1, the first set of classifiers (i.e. the gaze-based classifiers) correspond to the database of hand-eye coordination and gaze behavior models whereas the second set of classifiers (i.e. the sketch-based classifiers) correspond to the database of pointer behavior models. Lastly, the third set of classifiers combine the former two sets via classifier-level fusion technique.

Gaze-based classifiers depend solely on gaze-based features that (1) capture the dynamic aspects of hand-eye coordination behavior, and (2) quantify how the eye gaze data is structured in terms of saccades and fixations. Along with these features, we also feed the task type information to the classifiers. When training the classifiers, we use Gaussian mixture models (GMMs) which are often used in biometric systems due to their capability of representing a large class of sample distributions [64]. Individual mixtures of Gaussians are fitted for both the *target* and *outlier* data (having K_t and K_o Gaussians, respectively). Note that *target* data belongs to the user in consideration whereas *outlier* data belongs to the remaining users in our database. K_t and K_o are automatically estimated by comparing multiple models with varying numbers of components according to Bayes Information Criterion (BIC) statistic. BIC favors goodness-of-fit and parsimony – more complex models (with high numbers of estimated parameters) are penalized. Allocating a Gaussian per user and per task, we set the upper limits of K_t and K_o to 5 and 45 Gaussians, respectively.

Sketch-based classifiers summarize the image-based properties of the pointer data. Similarly, along with this information, we also feed the task type information to the classifiers. IDM Features have three free feature extraction parameters as k (kernel size), σ (smoothing factor), and r (resampling parameter). We set these parameters in accordance with the optimum values reported by Tümen et al. [78]. IDM feature vector is of size 720. Due to high dimensionality, it is not possible to use GMM classifiers with our sketch-based feature representation. Therefore, we use Support Vector Machines (SVM) binary classifiers with a Gaussian radial basis function

(RBF) kernel based on their established success in sketch recognition [79].

For building the combined classifiers via classifier-level fusion technique, we train a probabilistic GMM with the gaze-based features and a probabilistic SVM with the sketch-based features. The output of each probabilistic model is a single value representing the likelihood of the input sample belonging to the user in consideration. We then use the outputs of these two probabilistic models to train a third GMM.

5.2 Evaluation

During evaluation, we utilize our entire database of gaze and pointer data, which overall consists of 1500 task instances. For training and testing the binary classifiers, data of the user in consideration is fed to the binary classifier as data belonging to the *target* class whereas data of the remaining 9 users are fed to the binary classifier as data belonging to the *outlier* class. Data of both the *target* and *outlier* classes are further split into 5 folds randomly, but making sure that task instances are uniformly separated into each fold with respect to task type and scale. Out of these 5 folds, 4 are reserved for training the binary classifier whereas the remaining fold is reserved for testing the classifier. More specifically, for each binary classifier 1200 task instances (120 from *target* class and 1080 from *outlier* class) are used for training and 300 task instances (30 from *target* class and 270 from *outlier* class) are used for testing purposes. This process is repeated for each random split in a round-robin fashion such that each of the 5 folds is used exactly once for testing. For training and testing the binary classifiers, we use an open-source toolbox specialized in the research of one-class classification developed by Tax [77].

To evaluate the performance of the binary classifiers, we use area under the ROC curve (AUC) measure and equal error rate (EER). AUC measure computes the probability that a classifier will rank a randomly chosen positive instance (i.e. an authorized user) higher than a randomly chosen negative one (i.e. an intruder) (assuming positive ranks higher than negative). EER corresponds to the point where false acceptance rate is equal to false rejection rate, hence it represents a sort of steady state for the

evaluated classifier.

Gaze-based classifiers yielded a mean AUC score of $85.71 \pm 3.5\%$, and a mean EER of $21.75 \pm 4.05\%$ (Fig. 5.5, top row). A one-way ANOVA was conducted, however no significant differences were found among users with respect to AUC scores ($p = 0.319$) or EERs ($p = 0.12$). Peak performance was achieved for a random fold of the second user, with 97.36% AUC score and 7.45% EER (Fig. 5.6, top row). On the other hand, sketch-based classifiers yielded a mean AUC score of $63.11 \pm 6.01\%$, and a mean EER of $39.86 \pm 5.17\%$ (Fig. 5.5, bottom row). A one-way ANOVA was conducted, however no significant differences were found among users with respect to AUC scores ($p = 0.387$) or EERs ($p = 0.487$). Peak performance was achieved for a random fold of the ninth user, with 74.85% AUC score and 30.42% EER (Fig. 5.6, bottom row).

We conducted a paired-samples t-test to compare AUC scores in gaze-based binary classifier and sketch-based binary classifier conditions. There was a significant difference in AUC scores for these two conditions; $t(49) = 20.902, p = 0.000$. We conducted another paired-sampled t-test to compare EERs in gaze-based binary classifier and sketch-based binary classifier conditions. There was again a significant difference in EERs for these two conditions; $t(49) = -18.367, p = 0.000$. These results collectively suggest that the gaze-based feature representation is significantly better in capturing the richness and complexity of biometric user input when compared to a pointer-based feature representation that has been shown to work well for hand-drawn sketch data.

Following the individual performance tests, we conducted tests to explore whether gaze-based and sketch-based feature representations can be combined to improve performance of our gaze-based biometric authentication system. Combined classifiers yielded a mean AUC score of $83.93 \pm 5.06\%$, and a mean EER of $22.39 \pm 5.28\%$. A one-way ANOVA was conducted, however no significant differences were found among users with respect to AUC scores ($p = 0.223$) or EERs ($p = 0.171$). These results suggest that fusing the gaze-based and sketch-based feature representations did not yield an overall improved performance. However, if we look more closely it can be observed that some users have comparably more distinctive pointer behaviors. These users

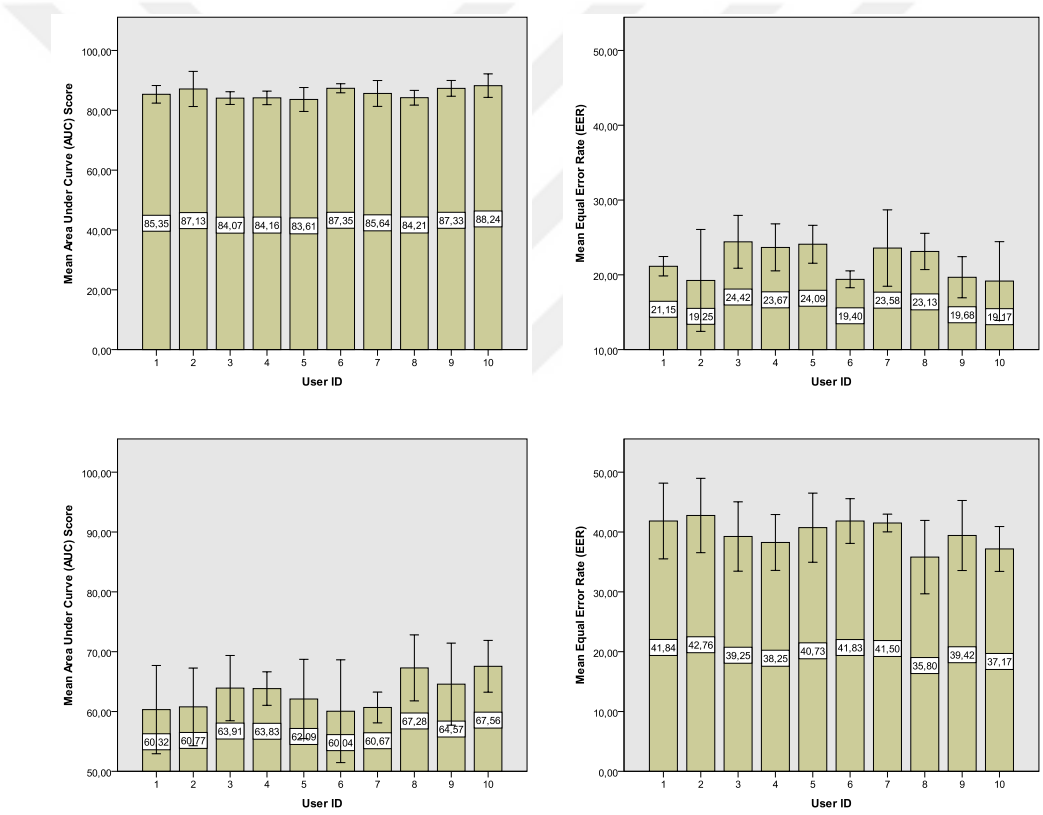


Figure 5.5: Performance of our gaze-based (top row) and sketch-based (bottom row) binary classifiers in terms of AUC score and EER, plotted for each user. Error bars indicate ±1 standard deviation.

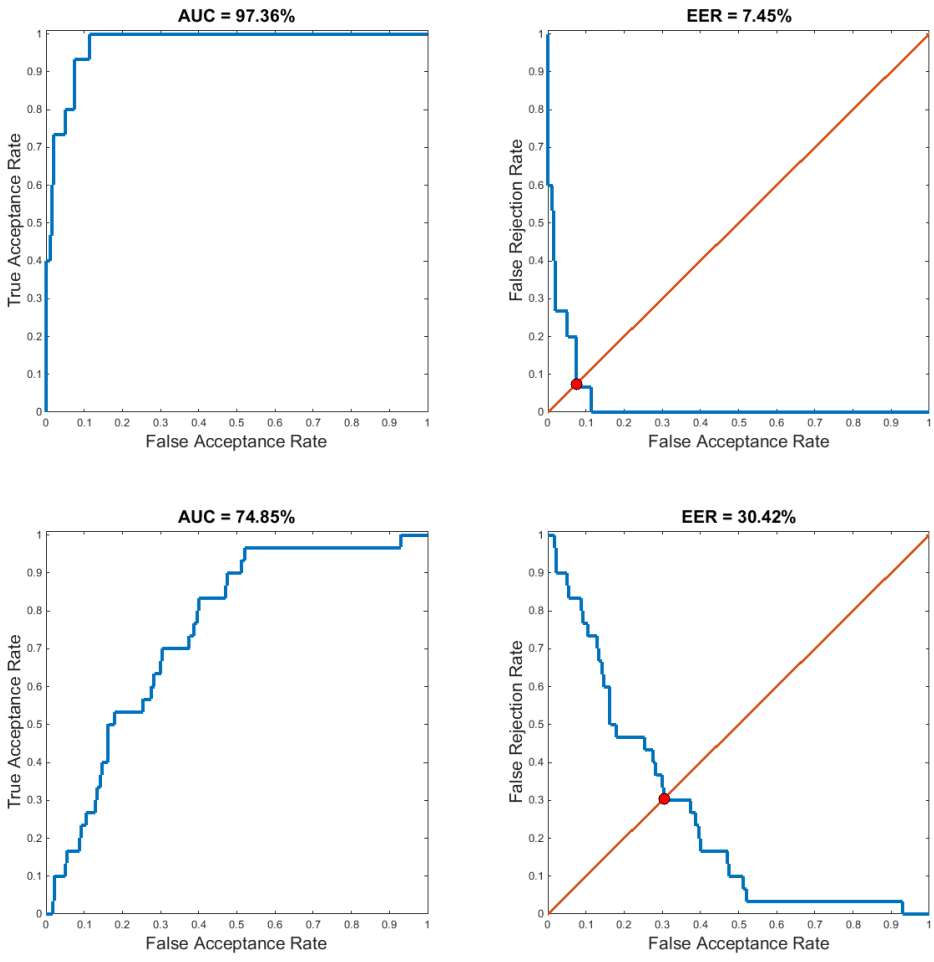


Figure 5.6: Peak performances of our gaze-based (top row) and sketch-based (bottom-row) binary classifiers in terms of AUC score and EER.

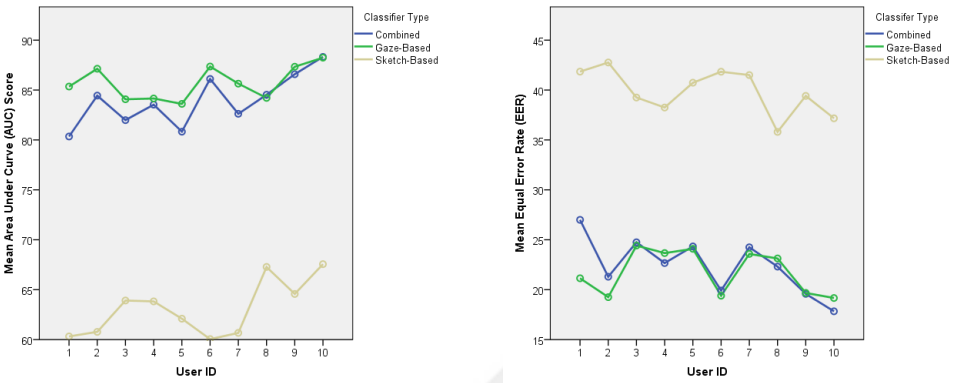


Figure 5.7: User 8 and User 10 have higher sketch-based AUC scores and lower sketch-based EERs compared to other users. Expectedly, these users have improved performances with the combined classifiers compared to both the gaze-based and sketch-based classifiers.

are marked with higher sketch-based AUC scores and lower sketch-based EERs than the overall population (Fig. 5.7). For users with comparably more distinctive pointer behaviors, fusing gaze-based and pointer-based features further improves the robustness of our authentication system. Visualization of an exemplary decision boundary for authenticating such users provides additional evidence to support this argument (Fig. 5.8).

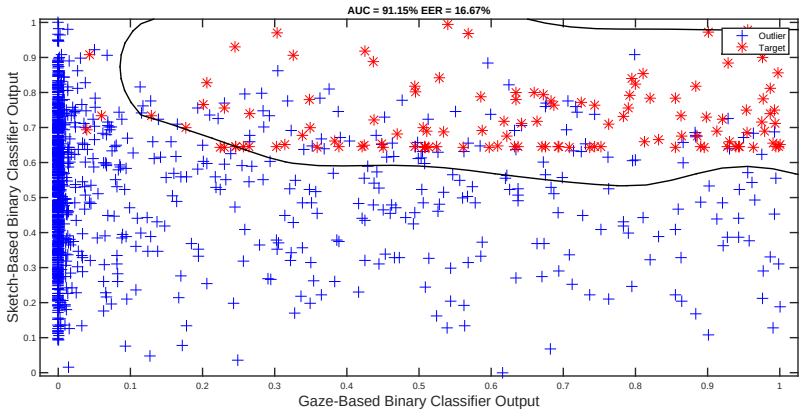


Figure 5.8: Decision boundary for a random fold of User 10. Red crosses accumulated around the upper portion of the image mark the *target* data instances. Blue plus signs scattered around the image mark the *outlier* data instances. Decision boundary separates the *target* instances from the *outlier* instances. Horizontal and vertial axes represent the likelihood values output by the gaze-based and sketch-based binary classifiers, respectively. Sketch-based classifier plays a significant role in determining the decision boundary since the likelihood values of the *target* class output by the gaze-based classifier do not fall into a specific range, and span nearly the whole domain instead.

Chapter 6

CONTRIBUTIONS

We have proposed a gaze-based virtual task prediction system to alleviate dependence on explicit mode switching in pen-based systems. Our system infers intended user actions by monitoring and analyzing eye gaze movements that users naturally exhibit during pen-based user interaction. More specifically, our system successfully discriminates between frequently employed pen-based virtual manipulation commands: *drag*, *maximize*, *minimize*, and *scroll*. In addition, our system differentiates between the intention to sketch and the intention to issue a command. We believe that predicting the mode of interaction will eventually allow us to build systems that save users the trouble of mode switching during basic interaction tasks. Our first contribution is a carefully compiled multimodal dataset that consists of eye gaze and pen input collected from participants completing various virtual interaction tasks. Our second contribution is a novel gaze-based feature representation, which is rooted in our understanding of human perception and gaze behavior. Our feature representation is neither subject- nor interface-specific, and performs better than common, well-established sketch recognition feature representations in the literature. Our third contribution is a novel gaze-based task prediction system based on this feature representation that can generalize to variations in task type and scale. The prediction results that we report are substantially better than existing work in the literature that attempt multi-class intention prediction as we do. Furthermore, we do not require defining application and interface specific areas of interests.

We have also presented the first line of work that uses online feedbacks from a gaze-based task prediction model to build a user interface that dynamically adapts itself to user's spontaneous task-related intentions and goals. Since it is not yet possible

to train prediction models that can perform with 100% accuracy, we have proposed novel approaches to providing visual feedback in the presence of uncertainty. From another point of view, we have closed the loop between the user and the prediction system by feeding highly accurate but imperfect predictions made by the prediction system to the user via appropriate visualizations of the user interface. Our novel approaches for visualizing uncertainty, namely *simultaneous visualization* and *adaptive transparency*, have been realized via wizard-based user interfaces and different flavors of predictive user interfaces. To assess the performance and preferrability of our interfaces, we have conducted a thorough usability study with 19 participants and 5 frequently employed virtual interaction tasks. Among these interfaces, *after-the-fact predictive UI* and *subtle real-time predictive UI* stand out as the best candidates for solving the uncertainty visualization challenge. Both interfaces are able to visualize user's task-related intentions and goals in the presence of uncertainty, and without significantly affecting user behavior and inhibiting performance of the underlying prediction systems. Moreover, the latter has comparable usability and perceived task load to WIMP-based user interfaces. Furthermore, we have offered a method to predict which predictive user interface will be more suitable for each user in terms of system performance. Personalization boosts system performance and provides users with the more preferred visualization approach.

We have also proposed a biometric authentication system that is based on natural, unconscious, and therefore inherently inimitable gaze behavior. Via our authentication system, a user can implicity communicate his/her identity to any gaze-enabled pointer-based device including, but not limited to, increasingly prominent pen-based mobile devices. To this end, we contribute a novel set of authentication tasks that involve manipulating a virtual object in familiar ways such as by dragging, resizing, scrolling etc. Short and simple, these tasks serve to provide general-purpose methods for everyday identification activities. To gain authorized access via these tasks, the user *must* manipulate the object correctly, and the user's precise patterns of hand-eye coordination, gaze, and pointer behaviors while performing the related task *should*

also confirm with the corresponding authentication models in our database. Our authentication system is fairly robust, with a mean AUC score of $85.71 \pm 3.5\%$, and a mean EER of $21.75 \pm 4.05\%$. Since our authentication system depends mostly upon behavioral characteristics of the eye rather than physical aspects of the oculomotor system, the importance of this robustness is further emphasized. With this work, we believe we have extended the research domain focusing on gaze-based behavioral biometrics. To further demonstrate the potential of using gaze-based features for biometric authentication, we have shown that our gaze-based feature representation is significantly better in capturing the richness and complexity of biometric user input when compared to a common, well-established pointer-based feature representation in the literature. For users with comparably more distinctive pointer behaviors, fusing gaze-based and pointer-based features further improves the robustness of our authentication system.

Chapter 7

FUTURE WORK AND CONCLUDING REMARKS

In the light of promising findings reported in this thesis, we envision a number of long-term research directions to explore for our gaze-based virtual task prediction system, our gaze-based intelligent user interface, and our gaze-based biometric authentication system.

Below is a tentative list of the long-term extensions that we intend to focus on for our gaze-based virtual task prediction system:

- Developing an exhaustive taxonomy of pen-based virtual interaction tasks. Using WordNet, we have already rounded up a list of approximately 200 actions. We plan to categorize these actions with respect to user's major high-level interaction goal into four groups as translation, manipulation, selection, and search.
- Conducting experiments to see if our prediction system can successfully recognize other virtual tasks. This may involve defining and training classifiers for possible subtypes of the *free-form drawing* task such as handwriting, drawing, selection (via underlining, circling, or pointing), and deletion (via erasing or scratching).
- Exploring if variants of a particular virtual task can be discriminated. For example, it is conceivable that a minimization task where the target size is set in reference to another virtual object may result in different stylus-gaze behavior compared to the case without a reference object. This may involve building a finer taxonomy of virtual tasks (e.g. drag with/without a target, minimize with/without a reference, etc.), and extending the feature representation to handle these finer distinctions.[8].

[8]In fact, we believe existing categorization of virtual tasks is rather coarse, and there is a pressing

- Conducting experiments to verify whether our task prediction system or a similar system inspired by our current findings generalizes well to other pointer-based user interfaces that accept stylus, finger, or mouse input rather than being limited to pen-based user interfaces only [59, 60].
- In the case that the technical limitations of the Tobii X120 eye tracker (i.e. the requirement to place the eye tracker below the interaction screen) are lifted, collecting new data in a direct input configuration where the input and display are collocated and reevaluating the effectiveness of our novel gaze-based feature representation in predicting virtual interaction tasks.

Below is a tentative list of the long-term extensions that we intend to focus on for our gaze-based intelligent user interface:

- Until we can build prediction models that can perform with 100% accuracy, we need to find a way to handle prediction errors. Although we have demonstrated that there is no significant effect of absence/presence of prediction errors on accuracy scores in our context, it is possible that users might confuse system errors with user-induced errors and diverge from natural gaze behavior in an effort to avoid them. In turn, this divergence will conceivably reduce the quality of the user's experience with the interface as well as the accuracy of our prediction systems that assume natural user behavior. In consequence, several questions remain to be addressed with respect to detecting and recovering from prediction errors: What will be the degree of initiative on the system's side – "will the system act, offer to act, ask if it should act, or merely indicate that it can act?" [38] How can we detect prediction errors? Will it be possible for users to correct prediction errors by overriding? How can we design transitions between implicit and explicit interaction, so that users can interrupt or stop a proactive system action? How can we establish shared understanding between

need to construct a fine grained taxonomy that highlights the differences between various flavors of these tasks.

the user and the system without interrupting the interaction flow? Formal user studies will be needed to obtain definitive answers to such questions.

- One remaining issue concerns mismatch between training and testing conditions of our gaze-based task prediction models. The mismatch is firstly due to the fact that our models were evaluated using a different group of participants than the one which provided the multimodal data for training them. In our future research we intend to concentrate on training the underlying prediction models using only the current user's data or data collected from users who exhibit similar hand-eye coordination behaviors to the current user's. The mismatch is secondly due to the fact that our models were trained with offline interaction data that do not involve user interface adaptations or prediction errors. Nevertheless, our models were tested in an online setting. Therefore, further research is required to investigate the performance of new prediction models trained using multimodal data collected during the usability study presented in this thesis. We believe that alleviating the mismatch problem will further boost the performance of our prediction systems.
- Another issue concerns *compatibility* prediction. We predict a user's *compatibility* with our gaze-based task prediction systems based on his/her performance in *wizard UI*. *Wizard UI* is designed to resemble as closely as possible the WIMP-based user interfaces that users are familiar with. Further study into predicting a user's *compatibility* based on his/her natural gaze behaviors during interaction with prominent browsers/operating systems (e.g. while the user is freely browsing the web, or organizing digital photo albums) would be of interest.

Below is a tentative list of the long-term extensions that we intend to focus on for our gaze-based biometric authentication system:

- On the basis of the promising findings presented in this thesis, the next stage of our research will entail a formal user study to evaluate our gaze-based authentication system in realistic usage scenarios. In addition to analyzing real-world

sensitivity and specificity of our system, the planned user study will include metrics for assessing usability. We intend to integrate our authentication system to a range of gaze-enabled mobile devices with varying sizes to test the scale-invariance of our system. We also intend to conduct evaluations on different days to test the consistency of our system.

- Clearly, further research will be needed to extend our novel set of gaze-based biometric authentication tasks with new tasks, or enhanced versions of existing tasks. Envisioned enhancements aim to improve security by giving users the freedom to fine-tune our tasks, thus making our tasks more customized and less discoverable. One such enhancement will allow users to choose the set of objects relevant for authentication among multiple objects presented on the device display. Another will allow them to determine the initial and final positions of the virtual object to be manipulated, or to define more complex paths with via points for the virtual object to follow during manipulation. More experiments will be needed to verify whether such enhancements improve the robustness of our authentication system, and whether the enhanced versions of our tasks are still familiar to users, easy/quick to perform, and able to consistently elicit characteristic gaze behavior from users for biometric authentication.

Below is a tentative list of the more general research questions we aim to address in the field of gaze-based interaction:

- Exploring the suitability of our feature representation scheme to aid the recognition and segmentation of sketches (e.g. URL diagrams, circuit diagrams). It is conceivable that conceptually different subtasks of sketching - such as drawing objects, connectors, arrows or producing handwritten annotations - may each have distinct gaze-stylus interactions, which might be captured by extending feature representations introduced in this thesis.
- Exploring the suitability of using eye gaze modality for predicting interruptibility. The concept of calm technology [82] is gaining importance. People often

use eye gaze when profiling each other's availability. Can we utilize user's gaze behavior along with information about his/her computer activity to determine whether the user is available for interruption?

- Exploring the suitability of using eye gaze modality for predicting "incorrect auto-corrections". Screen size limitations and the absence of a mouse make high-precision pointing impossible in mobile devices (also known as the fat finger problem). As a result, typographical errors are very common. These errors are automatically corrected by various AutoCorrect mechanisms to help users save time. However, it's often the case that the errors are intentional or a result of the AutoCorrect mechanism not being able to recognize the input text. Can we utilize user's gaze behavior to determine whether the seemingly unintentional typographical error is in fact intentional?

Appendix A

RELATED GAZE-BASED TASK PREDICTION APPROACHES

Paper Title	Primary Author	Year	Analyze/Predict	Daily/Virtual	Tasks
In what ways do eye movements contribute to everyday activities?	Micheal F. Land	2001	Analyze	Daily	Tea making Sandwich making
A robust algorithm for reading detection	Christopher S. Campbell	2001	Predict	Virtual	Reading Skimming Scanning
Understanding human behaviors based on eye-head-hand coordination	Chen Yu	2002	Predict	Daily	Unscrewing a jar Stapling a letter Pouring water
Learning to recognize human action sequences	Chen Yu	2002	Predict	Daily	Stapling a letter
Using eye gaze patterns to identify user tasks	Shamsi Tamara Iqbal	2004	Analyze	Virtual	Reading comprehension Mathematical reasoning Search Object manipulation
Eye movements in natural behavior	Mary Hayhoe	2005	Analyze	Daily	Everyday visually guided behaviors
Eye and pen: a new device for studying reading during writing	Denis Alamargot	2006	Analyze	Virtual	Reading Writing
Human gaze behavior during action execution and observation	Benno Gesierich	2008	Analyze	Virtual	Action execution Action observation
Recognizing behavior in hand-eye coordination patterns	Weilie Yi	2009	Analyze	Daily	10 subtasks of a sandwich making task
Multimodal integration of natural gaze behavior for intention recognition during object manipulation	Thomas Bader	2009	Predict	Virtual	Object manipulation
Eye movement analysis for activity recognition	Andreas Bulling	2009	Predict	Daily	Copy Read Write Video Browse Null
What's in the eyes for context-awareness?	Andreas Bulling	2011	Predict	Daily	- Reading or not reading - Copy, read, write, video, browse, null - Visual memory (Familiar/unfamiliar images)
Activity recognition using eye-gaze movements and traditional interactions	François Courtemanche	2011	Predict	Virtual	Subtasks of Google Analytics and eLearning tasks
Learning to recognize daily actions using gaze	Alireza Fathi	2012	Predict	Daily	Meal preparation
What do you want to do next: a novel approach for intent prediction in gaze-based interaction	Roman Bednarik	2012	Predict	Virtual	Issue a command or not
Coupling eye-motion and ego-motion features for first-person activity recognition	Keisuke Ogaki	2012	Predict	Daily	Copy Read Write Video Browse Null
User-adaptive information visualization – Using eye gaze data to infer visualization tasks and user cognitive abilities	Ben Steichen	2013	Predict	Virtual	Retrieve value Filter Compute derived value Find extremum Sort

Appendix B

PSEUDOCODES FOR CHARACTERISTIC CURVE EXTRACTION ALGORITHMS

Algorithm 1 Algorithm for building the instantaneous sketch-gaze distance curves.

Input: Gaze data G_i and sketch data P_i for all task instances of the input task and scale

Output: Instantaneous sketch-gaze distance curves $D_i^{'}$

1: **for all** Task instances i **do**
2: $D_i \leftarrow |G_i - P_i|$
3: $D_i^{'} \leftarrow smooth(D_i)$
4: **end for**

Algorithm 2 Algorithm for forming the similarity matrix.

Input: $D_i^{'}$ for all task instances of the input task and scale

Output: Similarity matrix S

1: **for all** Task instance pairs i, j **do**
2: $S_{ij} \leftarrow dtw\_distance(D_i^{'}, D_j^{'})$ ▷ *dtw_distance* method computes the similarity of two given curves.
3: **end for**

Algorithm 3 Algorithm for extracting the characteristic curve(s).

Input: Similarity matrix S of the input task and scale
Output: An array of characteristic curve(s)

1: $clusterArray \leftarrow linkage(S)$
2: **for all** clusters C_i in $clusterArray$ with at least 10 task instances **do**
3: initialize $w_j \leftarrow 1$ for all members j of cluster C_i
4: **while** C_i contains multiple curves **do**
5: find the most similar curve pair $(first, second)$ in the cluster
6: warp the first curve with respect to the second curve to get $firstWarped$
7: warp the second curve with respect to the first curve to get $secondWarped$
8: $firstWeight \leftarrow \frac{w_{first}}{w_{first}+weight_{second}}$ and $secondWeight \leftarrow \frac{w_{second}}{w_{first}+w_{second}}$
9: take a weighted average of the warped curves to get $newCurve$
10: $w_{newCurve} \leftarrow w_{first} + w_{second}$
11: replace the warped curves in the cluster with the newly computed curve
12: **end while**
13: add the final $newCurve$ to the array of characteristic curves
14: **end for**

Appendix C

QUESTIONNAIRE USED IN THE USABILITY STUDY

C.1 English Translation

Dear Participant,

The answers you give to this questionnaire will be used for my doctoral thesis studies. No information will be requested from you that can be used for revealing your identity. The answers you give to this questionnaire will not be shared with anyone and will not be used for a purpose other than that indicated.

Please be sure to answer all questions completely.

Çağla Çığ

Age:

Sex:

Level of Education:

Eye Color:

Do you have any vision problems?

Which hand do you write with?

How experienced are you with tablet devices?

Never used them		Use them daily		Developed software using these devices
1	2	3	4	5

How experienced are you with pen-based tablet devices?

Never used them		Use them daily		Developed software using these devices
1	2	3	4	5

How experienced are you with eye tracking devices?

Never heard of them		Know their functionalities		Developed software using these devices
1	2	3	4	5

During this experiment, you have essentially used **3 different user interfaces.**

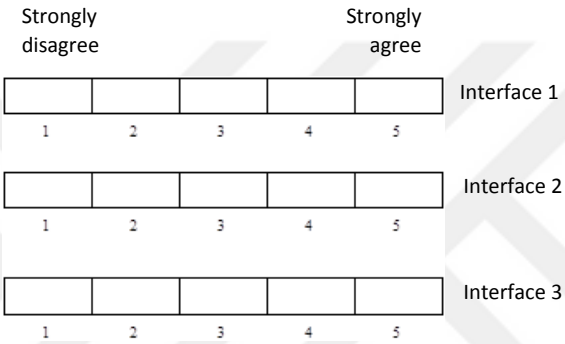
In Interface 1, there were feedbacks regarding your current task only.

In Interface 2, there were feedbacks regarding both your current task and other tasks.

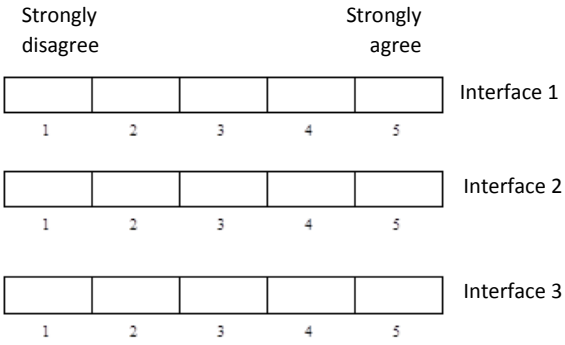
In Interface 3, there were feedbacks regarding both your current task and other tasks. Moreover, there was an intelligent system working in the background. This system guessed your current task and adjusted the transparency levels of irrelevant tasks accordingly.

Please answer the following questions for these 3 interfaces.

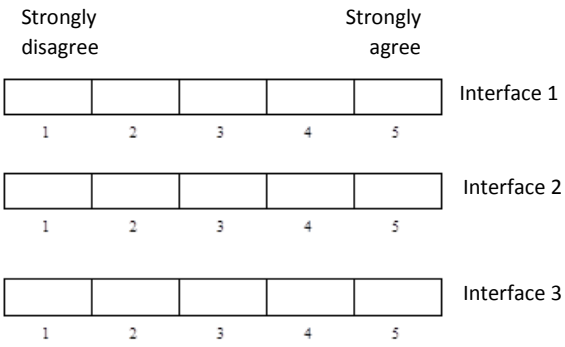
1. **I thought the system was easy to use.**



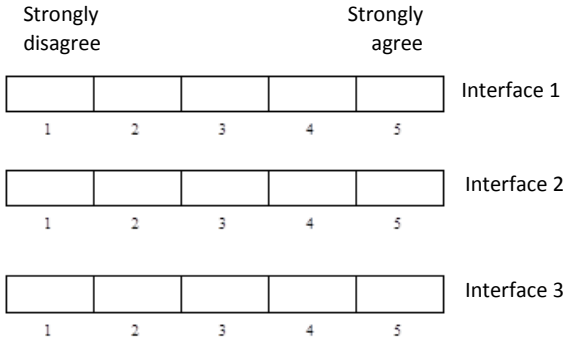
2. **I found the system unnecessarily complex.**



3. **I would imagine that most people would learn to use this system very quickly.**

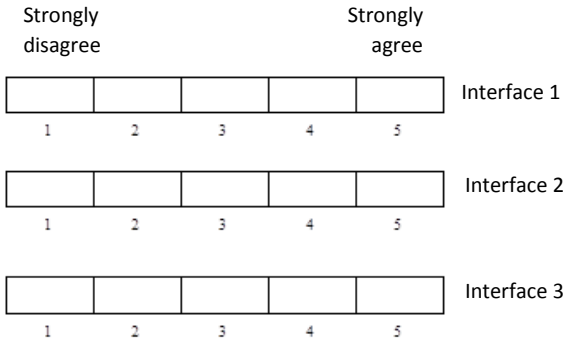


4. I thought there was too much inconsistency in this system.

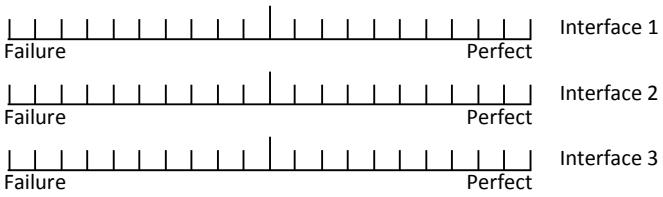


5. I felt very confident using the system.

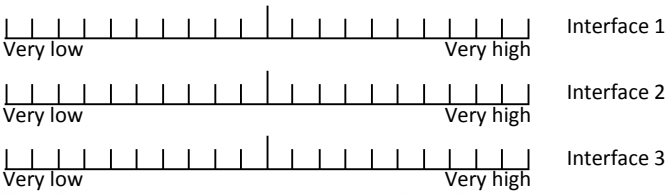
6. I needed to learn a lot of things before I could get going with this system.



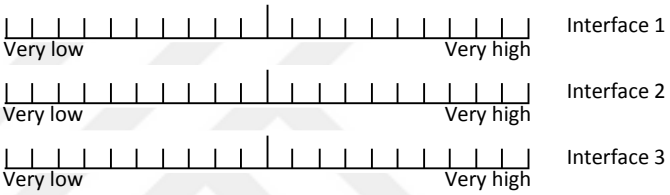
1. How successful were you in accomplishing what you were asked to do?



2. How hard did you have to work to accomplish your level of performance?



3. How insecure, discouraged, irritated, stressed, and annoyed were you?



Additional Comments:

C.2 Original Turkish Version

Değerli Katılımcı,

Bu ankete vereceğiniz yanıtlar, yazmakta olduğum doktora tezi için bilgi toplamak amaçlıdır. Kimliğinizi açığa çıkaracak hiçbir bilginin talep edilmediği bu ankete vereceğiniz yanıtların kimseyle paylaşılmayacağından ve başka bir amaçla kullanılmayacağından emin olabilirsiniz.

Lütfen tüm soruları eksiksiz bir şekilde yanıtladığınızdan emin olunuz.

Çağla Çığ

Yaşınız:

Cinsiyetiniz:

Öğrenim Durumunuz:

Göz Renginiz:

Görme bozukluğunuz var mı?

Yazarken hangi elinizi kullanıyorsunuz?

Dokunmatik ekranlı cihazlar üzerine ne kadar tecrübelisiniz?

Hiç kullanmadım		Günlük kullanıyorum		Bu cihazlar ile uygulama geliştirdim
1	2	3	4	5

Kalemle kullanılan dokunmatik ekranlı cihazlar üzerine ne kadar tecrübelisiniz?

Hiç kullanmadım		Günlük kullanıyorum		Bu cihazlar ile uygulama geliştirdim
1	2	3	4	5

Göz takip cihazları üzerine ne kadar tecrübelisiniz?

Hiç duymadım		İşlevlerini biliyorum		Bu cihazlar ile uygulama geliştirdim
1	2	3	4	5

Deneyde temelde **3 farklı ara yüz** kullandınız.

Arayüz 1'de sadece gerçekleştirmekte olduğunuz göreve ait geri bildirimler vardı.

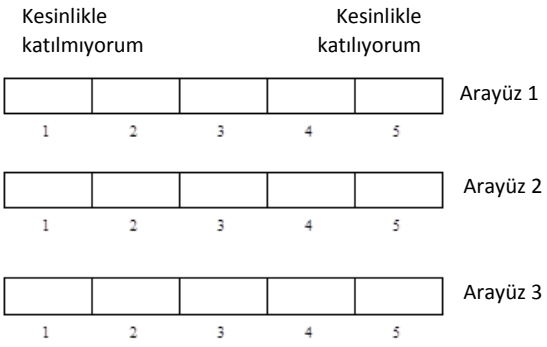
Arayüz 2'de hem gerçekleştirmekte olduğunuz göreve ait hem de farklı görevlere ait geri bildirimler vardı.

Arayüz 3'te de hem gerçekleştirmekte olduğunuz göreve ait hem de farklı görevlere ait geri bildirimler vardı. Buna ek olarak arka planda bir akıllı sistem çalışıyordu. Bu sistem, sizin o sırada yapmakta olduğunuz görevi tahmin edip ilgisiz bulduğu görevlerin şeffaflık/saydamlık seviyelerini uygun şekilde ayarlıyordu.

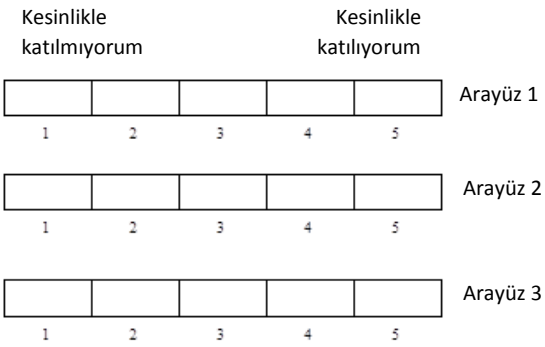
Lütfen aşağıdaki soruları bu 3 ara yüz için cevaplandırınız.

1. **Sistemin kolay kullanıldığını düşündüm.**

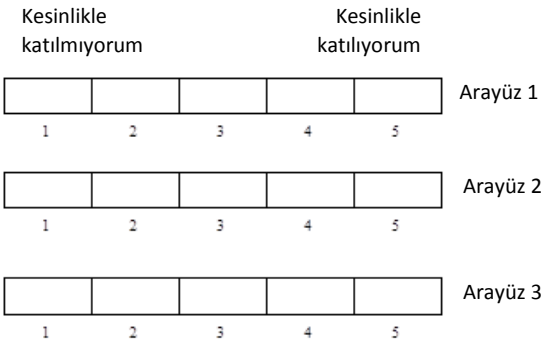
2. **Sistemi gereksiz bir şekilde karmaşık buldum.**



3. **Birçok insanın bu sistemi hızlı bir şekilde kullanabileceğini düşünüyorum.**

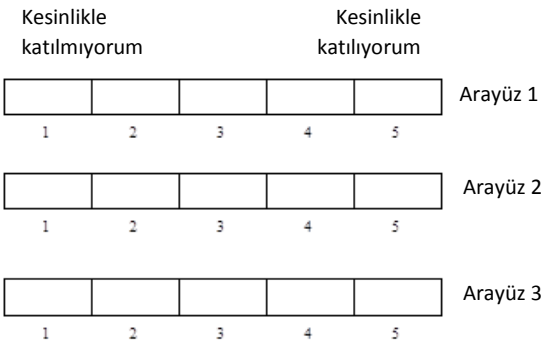


4. Sistemde çok fazla tutarsızlık olduğunu düşündüm.

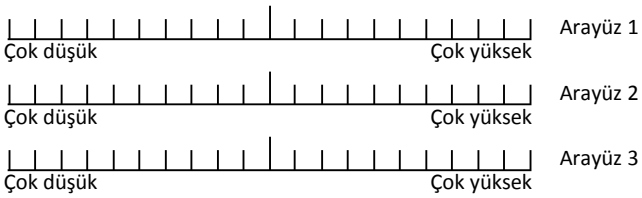


5. Sistemi kullanırken kendimden emindim.

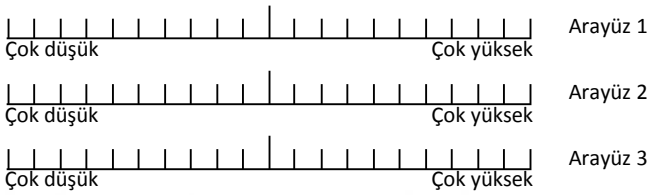
6. Sistemi kullanmadan önce birçok şey öğrenmem gerekti.



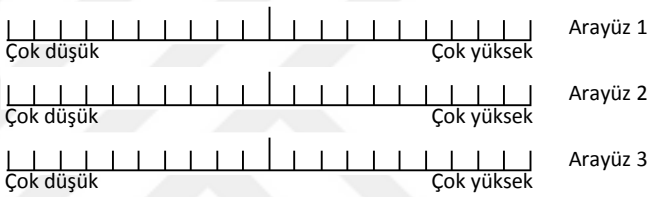
1. Size verilen görevleri ne ölçüde başarıyla yerine getirdiniz?



2. Size verilen görevleri yerine getirebilmek için ne kadar çaba sarf etmeniz gerekti?



3. Size verilen görevleri yerine getirirken kendinizi ne ölçüde güvensiz, irrite olmuş ve gergin hissettiniz?



Ek Yorumlar:

BIBLIOGRAPHY

[1] Adler, F. H. (1959). *Physiology of the eye: clinical application*. Mosby.

[2] Alamargot, D., Chesnet, D., Dansac, C., and Ros, C. (2006). Eye and pen: a new device for studying reading during writing. *Behav. Res. Methods*, 38(2):287–299.

[3] Bader, T., Vogelgesang, M., and Klaus, E. (2009). Multimodal integration of natural gaze behavior for intention recognition during object manipulation. In *Proceedings of the Eleventh International Conference on Multimodal Interfaces*, pages 199–206, New York, NY, USA. ACM.

[4] Ballard, D. H., Hayhoe, M. M., Li, F., Whitehead, S. D., Frisby, J. P., Taylor, J. G., and Fisher, R. B. (1992). Hand-eye coordination during sequential tasks [and discussion]. *Philos. Trans. Biol. Sci.*, 337(1281):331–339.

[5] Bednarik, R., Kinnunen, T., Mihaila, A., and Fränti, P. (2005). Eye-movements as a biometric. In *Image Analysis*, pages 780–789. Springer Berlin Heidelberg.

[6] Bednarik, R., Vrzakova, H., and Hradis, M. (2012). What do you want to do next: a novel approach for intent prediction in gaze-based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 83–90, New York, NY, USA. ACM.

[7] Biedert, R., Frank, M., Martinovic, I., and Song, D. (2012). Stimuli for gaze based intrusion detection. In *Future Information Technology, Application, and Service*, pages 757–763. Springer Science & Business Media.

[8] Brooke, J. (1996). Sus - a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

[9] Bulling, A., Roggen, D., and Tröster, G. (2011). What's in the eyes for context-awareness? *Pervasive Computing, IEEE*, 10(2):48–57.

[10] Bulling, A., Ward, J. A., Gellersen, H., and Tröster, G. (2009). Eye movement analysis for activity recognition. In *Proceedings of the Eleventh International Conference on Ubiquitous Computing*, pages 41–50, New York, NY, USA. ACM.

[11] Campbell, C. S. and Maglio, P. P. (2001). A robust algorithm for reading detection. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, pages 1–7, New York, NY, USA. ACM.

[12] Cantoni, V., Galdi, C., Nappi, M., Porta, M., and Riccio, D. (2015). Gant: gaze analysis technique for human identification. *Pattern Recogn.*, 48(4):1027–1038.

[13] Carenini, G., Conati, C., Hoque, E., Steichen, B., Toker, D., and Enns, J. (2014). Highlighting interventions and user differences: informing adaptive information visualization support. In *Proceedings of the Thirty-second Annual ACM Conference on Human Factors in Computing Systems*, pages 1835–1844, New York, NY, USA. ACM.

[14] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Trans. on Intell. Syst. and Technol.*, 2(3):1–27.

[15] Chaudhri, I., Ording, B., Anzures, F. A., van Os, M., Lemay, S. O., Forstall, S., and Christie, G. (2011). Unlocking a device by performing gestures on an unlock image. US Patent 8,046,721.

[16] Çığ, Ç. and Sezgin, T. M. (2015a). Gaze-based prediction of pen-based virtual interaction tasks. *Int. J. Hum.-Comput. Stud.*, 73:91–106.

[17] Çığ, Ç. and Sezgin, T. M. (2015b). Real-time activity prediction: a gaze-based approach for early recognition of pen-based interaction tasks. In *Proceedings of the Twelfth Sketch-Based Interfaces and Modeling Symposium*, pages 59–65, Aire-la-Ville, Switzerland. Eurographics Association.

[18] Conati, C., Carenini, G., Hoque, E., Steichen, B., and Toker, D. (2014). Evaluating the impact of user characteristics and different layouts on an interactive visualization for decision making. *Comput. Graph. Forum*, 33(3):371–380.

[19] Courtemanche, F., Aïmeur, E., Dufresne, A., Najjar, M., and Mpondo, F. (2011). Activity recognition using eye-gaze movements and traditional interactions. *Interact. Comput.*, 23(3):202–213.

[20] Darwish, A. and Pasquier, M. (2013). Biometric identification using the dynamic features of the eyes. In *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6.

[21] de Xivry, J.-J. O. and Lefèvre, P. (2007). Saccades and pursuit: two outcomes of a single sensorimotor process. *J. Physiol.*, 584(1):11–23.

[22] DeBarr, D. (2006). Constrained dynamic time warping distance measure. http://www.mathworks.com/matlabcentral/fileexchange/12319-constrained-dynamic-time-warping-distance-measure/.

[23] Deravi, F. and Guness, S. P. (2011). Gaze trajectory as a biometric modality. In *Proceedings of the International Conference on Bio-Inspired Systems and Signal Processing*, pages 335–341.

[24] D'Mello, S., Olney, A., Williams, C., and Hays, P. (2012). Gaze tutor: a gaze-reactive intelligent tutoring system. *Int. J. Hum.-Comput. Stud.*, 70(5):377–398.

[25] Doggart, J. H. (1949). *Ocular signs in slit-lamp microscopy*. Henry Kimpton.

[26] Duchowski, A. T., Cournia, N., and Murphy, H. A. (2004). Gaze-contingent displays: a review. *Behav. Social Netw.*, 7(6):621–634.

[27] Fathi, A., Li, Y., and Rehg, J. M. (2012). Learning to recognize daily actions using gaze. In *Proceedings of the Twelfth European Conference on Computer Vision - Volume Part I*, pages 314–327, Berlin, Heidelberg. Springer-Verlag.

[28] Felty, T. (2004). Dynamic time warping. http://www.mathworks.com/matlabcentral/fileexchange/6516-dynamic-time-warping/.

[29] Forlines, C. and Balakrishnan, R. (2008). Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1563–1572, New York, NY, USA. ACM.

[30] Gesierich, B., Bruzzo, A., Ottoboni, G., and Finos, L. (2008). Human gaze behaviour during action execution and observation. *Acta Psychol.*, 128(2):324–330.

[31] Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): results of empirical and theoretical research. *Advances in psychology*, 52:139–183.

[32] Hayhoe, M. and Ballard, D. (2005). Eye movements in natural behavior. *Trends Cogn. Sci.*, 9(4):188–194.

[33] Holland, C. and Komogortsev, O. V. (2011). Biometric identification via eye movement scanpaths in reading. In *Proceedings of the International Joint Conference on Biometrics*, pages 1–8.

[34] Hyndman, R. J. (2004). Fast computation of cross-validation in linear models. http://robjhyndman.com/hyndsight/loocv-linear-models/.

[35] Iqbal, S. T. and Bailey, B. P. (2004). Using eye gaze patterns to identify user tasks. In *The Grace Hopper Celebration of Women in Computing*.

[36] James, G. M. (2007). Curve alignment by moments. *Ann. Appl. Stat.*, 1(2):480–501.

[37] Johansson, R. S., Westling, G., Bäckström, A., and Flanagan, J. R. (2001). Eye-hand coordination in object manipulation. *J. Neurosci.*, 21(17):6917–6932.

[38] Ju, W., Lee, B. A., and Klemmer, S. R. (2008). Range: exploring implicit interaction through electronic whiteboard design. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pages 17–26, New York, NY, USA. ACM.

[39] Kasprowski, P. and Ober, J. (2004). Eye movements in biometrics. In *Biometric Authentication*, pages 248–258. Springer Berlin Heidelberg.

[40] Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):489–497.

[41] Kinnunen, T., Sedlak, F., and Bednarik, R. (2010). Towards task-independent person authentication using eye movement signals. In *Proceedings of the Symposium on Eye-Tracking Research & Applications*, pages 187–190.

[42] Kneip, A. and Gasser, T. (1992). Statistical tools to analyze data representing a sample of curves. *Ann. Stat.*, 20(3):1266–1305.

[43] Komogortsev, O. V., Jayarathna, S., Aragon, C. R., and Mahmoud, M. (2010). Biometric identification via an oculomotor plant mathematical model. In *Proceedings of the Symposium on Eye-Tracking Research & Applications*, pages 57–60.

[44] Komogortsev, O. V., Karpov, A., Holland, C. D., and Proenca, H. P. (2012). Multimodal ocular biometrics approach: a feasibility study. In *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 209–216.

[45] Kumar, M., Paepcke, A., and Winograd, T. (2007). Eyepoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 421–430, New York, NY, USA. ACM.

[46] Land, M. F. and Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision Res.*, 41(25–26):3559–3565.

[47] Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. (2005). Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 461–470, New York, NY, USA. ACM.

[48] Liang, Z., Tan, F., and Chi, Z. (2012). Video-based biometric identification using eye tracking technique. In *Proceedings of the IEEE International Conference on Signal Processing, Communication and Computing*, pages 728–733.

[49] Meeker, M. and Wu, L. (2013). Internet trends d11 conference. http://www.kpcb.com/insights/2013-internet-trends/.

[50] Negulescu, M., Ruiz, J., and Lank, E. (2010). Exploring usability and learnability of mode inferencing in pen/tablet interfaces. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 87–94, Aire-la-Ville, Switzerland. Eurographics Association.

[51] Nielsen, J. (1993). Noncommand user interfaces. *Commun. ACM*, 36(4):83–99.

[52] Nisenson, M., Yariv, I., El-Yaniv, R., and Meir, R. (2003). Towards behaviometric security systems: learning to identify a typist. In *Knowledge Discovery in Databases: PKDD 2003*, pages 363–374. Springer Berlin Heidelberg.

[53] Norman, D. A. (1988). *The design of everyday things*. Basic Book.

[54] Nugrahaningsih, N. and Porta, M. (2014). Pupil size as a biometric trait. In *Biometric Authentication*, pages 222–233. Springer International Publishing.

[55] Ogaki, K., Kitani, K. M., Sugano, Y., and Sato, Y. (2012). Coupling eye-motion and ego-motion features for first-person activity recognition. In *Computer Vision and Pattern Recognition Workshops*, pages 1–7. IEEE.

[56] Okoe, M., Alam, S. S., and Jianu, R. (2014). A gaze-enabled graph visualization to improve graph reading tasks. *Comput. Graph. Forum*, 33(3):251–260.

[57] Ouyang, T. Y. and Davis, R. (2009). A visual approach to sketched symbol recognition. In *Proceedings of the Twenty-first International Joint Conference on Artifical Intelligence*, pages 1463–1468.

[58] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1226–1238.

[59] Pfeuffer, K., Alexander, J., Chong, M. K., and Gellersen, H. (2014). Gaze-touch: Combining gaze with multi-touch for interaction on the same surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 509–518, New York, NY, USA. ACM.

[60] Pfeuffer, K., Alexander, J., and Gellersen, H. (2016). Partially-indirect bimanual input with gaze, pen, and touch for pan, zoom, and ink interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2845–2856, New York, NY, USA. ACM.

[61] Plimmer, B. (2008). Experiences with digital pen, keyboard and mouse usability. *J. Multimodal User Interfaces*, 2(1):13–23.

[62] Ramsay, J. O. (2006). *Functional data analysis*. Wiley Online Library.

[63] Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *Q. J. Exp. Psychol.*, 62(8):1457–1506.

[64] Reynolds, D. (2015). Gaussian mixture models. *Encyclopedia of biometrics*, pages 827–832.

[65] Rigas, I., Economou, G., and Fotopoulos, S. (2012). Biometric identification based on the eye movements and graph matching techniques. *Pattern Recognition Letters*, 33(6):786–792.

[66] Roberts, C. (2007). Biometric attack vectors and defences. *Comput. Secur.*, 26(1):14–25.

[67] Rubine, D. (1991). Specifying gestures by example. *SIGGRAPH Comput. Graph.*, 25(4):329–337.

[68] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.*, 26(1):43–49.

[69] Schmidt, A. (2000). Implicit human computer interaction through context. *Personal Technologies*, 4(2-3):191–199.

[70] Sibert, J. L., Gokturk, M., and Lavine, R. A. (2000). The reading assistant: eye gaze triggered auditory prompting for reading remediation. In *Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software and Technology*, pages 101–107, San Diego, CA, USA. ACM.

[71] Silver, D. L. and Biggs, A. J. (2006). Keystroke and eye-tracking biometrics for user identification. In *Proceedings of the International Conference on Artificial Intelligence*, pages 344–348.

[72] Simon, C. and Goldstein, I. (1935). A new scientific method of identification. *New York State Journal of Medicine*, 35(18):901–906.

[73] Starker, I. and Bolt, R. A. (1990). A gaze-responsive self-disclosing display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3–10, New York, NY, USA. ACM.

[74] Steichen, B., Carenini, G., and Conati, C. (2013). User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the Eighteenth International Conference on Intelligent User Interfaces*, pages 317–328, New York, NY, USA. ACM.

[75] Steichen, B., Conati, C., and Carenini, G. (2014). Inferring visualization task properties, user performance, and user cognitive abilities from eye gaze data. *TiiS*, 4(2):1–29.

[76] Streit, M., Lex, A., Müller, H., and Schmalstieg, D. (2009). Gaze-based focus adaption in an information visualization system. In *IADIS International Conference Computer Graphics, Visualization, Computer Vision and Image Processing*, pages 303–307.

[77] Tax, D. (2015). Ddtools, the data description toolbox for matlab. http://prlab.tudelft.nl/david-tax/dd_tools.html. version 2.1.2.

[78] Tümen, R. S., Acer, M. E., and Sezgin, T. M. (2010). Feature extraction and classifier combination for image-based sketch recognition. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, pages 63–70, Aire-la-Ville, Switzerland. Eurographics Association.

[79] Ulaş, A., Yıldız, O. T., and Alpaydın, E. (2012). Cost-conscious comparison of supervised learning algorithms over multiple data sets. *Pattern Recogn.*, 45(4):1772 – 1781.

[80] Wang, H., Chignell, M., and Ishizuka, M. (2006). Empathic tutoring software agents using real-time eye tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 73–78, New York, NY, USA. ACM.

[81] Weiser, M. (1993). Ubiquitous computing. *IEEE Computer*, 26:71–72.

[82] Weiser, M. and Brown, J. S. (1996). Designing calm technology. *PowerGrid Journal*, v1.01.

[83] Yi, W. and Ballard, D. H. (2009). Recognizing behavior in hand-eye coordination patterns. *Int. J. Hum. Robot.*, 6(3):337–359.

[84] Yu, C. and Ballard, D. (2002a). Learning to recognize human action sequences. In *Proceeding of the Second International Conference on Development and Learning*, pages 28–33.

[85] Yu, C. and Ballard, D. H. (2002b). Understanding human behaviors based on eye-head-hand coordination. In *Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pages 611–619, London, UK. Springer-Verlag.

[86] Zhai, S., Morimoto, C., and Ihde, S. (1999). Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 246–253, New York, NY, USA. ACM.