WatchTrace: Design and Evaluation of an At-Your-Side Gesture Paradigm

by

Shaishav Siddhpuria

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2017

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes first-authored publication material from peer-reviewed conference proceedings published by the Association for Computing Machinery (ACM), along with the work and analysis carried out in conjunction with Keiko Katsuragawa, James Wallace, and my supervisor, Edward Lank. The contents of this thesis has been adapted, revised, and extended from the following conference publication:

- *Shaishav Siddhpuria, Keiko Katsuragawa, James R. Wallace, and Edward Lank. 2017. Exploring At-Your-Side Gestural Interaction for Ubiquitous Environments. In Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17). ACM, New York, NY, USA, 1111-1122. DOI: https://doi.org/10.1145/3064663.3064695*

My contribution includes: the implementation of the system, carrying out the complete labratory study, performing literature review, domain exploration, performing quantitative & qualitative analysis, and preparing the submission video for the system. I thank my co-authors for their continuous guidance, their help with writing & revising some of the paper content, and for aiding the creation of some of the experimental figures.

Abstract

In this thesis, we present the exploration and evaluation of a gesture interaction paradigm performed with arms at rest at the side of one's body. This gesture stance is informed persisting challenges in mid-air arm gesture interactions in relation to fatigue and social acceptability. The proposed arms-down posture reduces physical effort by minimizing the shoulder torque placed on the user. While this interaction posture has been previously explored, the gesture vocabulary in previous research has been small and limited. The design of this gesture interaction is motivated by the ability to provide rich and expressive input; the user performs gestures by moving the whole arm at the side of the body to create two-dimensional visual traces, as in hand-drawing in a bounded plane parallel to the ground. Within this space, we present the results of two studies that investigate the use of side-gesture input for interaction. First, we explore the users' mental model for using this interaction by conducting an elicitation study on a set of everyday tasks one would perform on a large display in public to semi-public contexts. The takeaway from this study presents the need for a dynamic and expressive set of gesture vocabulary including ideographic and alphanumeric gesture constructs that can be combined or chained together.

We then explore the feasibility of designing such a gesture recognition system using commodity hardware and recognition techniques, dubbed *WatchTrace*, which supports alphanumeric gestures of up to length three, providing a vibrant, dynamic, and feasible gestural vocabulary. Finally, we explore potential approaches to improve the recognition through the use of adaptive thresholds, n-best lists, and changing reject rates among other conventional techniques in the field of gesture classification.

Acknowledgements

I first would like to thank my supervisor Ed Lank, who has taught me, directly and indirectly, a lot of things about being a grad student, doing good research, being self-driven, and persisting through all kinds of difficulties. Ed, thank you for choosing to supervise me and putting a lot of faith in me to drive the kind of research I wanted to do. Despite being on a sabbatical for a good chunk of my studies (probably skiing in France), you prioritized my work at many points in time and made yourself available when I needed your guidance the most. I even got to spend a term in France because of you. Seriously, thanks a lot!

My two years in grad school would not have been so tolerable without my lab members, who offered advice, listened to my problems, and helped me along all sorts of professional and personal challenges. In the interest of not forgetting to list someone, I will keep it short and thank the entire HCI lab, its current and former members, and faculty Daniel Vogel and Edith Law for making me feel a welcome part of the lab. I will never forget the lab tours to on-campus Starbucks several times a day during paper deadlines.

The environment of the Waterloo HCI lab promotes such a high bar of excellence and collaboration. I'm lucky to wrap up my studies with two peer-reviewed publications (and a third in submission), in large parts due to my excellent, and highly patient, research collaborators: Keiko Katsuragawa, James Wallace, Mathieu Nancel, and Sylvain Malacria. I learned so much from you for what it takes to be a top-notch researcher.

Most importantly, I credit all my success to my parents and sister. I consider myself to be so, so lucky to have an unbelievable support system and freedom to pursue what I want to in life. It will have been twelve years since we moved as a family from India to Canada. Even though that came with countless adjustment, your support never wavered, and you allowed me the freedom to lead my own path. I couldn't ask for more.

Dedication

To Mom, Dad, and my sister Shailee.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Free-space interactions, such as gestures, involve using kinematic movements of various parts of the user's body such as the arm or the hand to interact and manipulate computer systems. Interaction with ubiquitous displays using free-space gestures has been an active area of research [58, 59, 2]. Free-space gesture techniques allow users to quickly and seamlessly interact with displays in their environment from a distance. Natural hand and arm gestures are most commonly used as a form of free-space gestural interaction to perform a natural movement as a metaphor for a task or an action. The technological advancement of ubiquitous displays, especially with the growth and miniaturization of the internet-of-things (IoT) devices [4, 48, 40] has also increased the space and complexity of actions available to the user.

Motion tracking devices and sensors have continually advanced its capability to identify gestural input both quickly and accurately. Commercial gesture recognition systems take various approaches for tracking and identifying motion gestures: camera-based computer vision techniques [45, 61], sensing electrical activity of the skeletal muscles through electromyography (EMG) [26], and measuring arm and hand kinetic force through the accelerometer and other sensors forming the inertial measurement unit (IMU) [83, 60, 17].

1.1 Motivation

Researchers have long tried to quantify the effortful nature of gesture-based communication with metrics such as *consumed endurance* [27], characterizing the effort one must

expend to support interaction. Noise is another challenge in free space interaction as it is difficult to infer user motions and distinguish between intended and unintended input in lieu of additional constraints placed in the system to guide or segment the input. Finally, sensing technology is challenging. Large scale gestures are easy to sense from afar, but small scale and subtle movements require spatial, size, or kinematic constraints. Direct-touch input, as an alternative, is less desirable in public spaces, considering issues such as occlusion and a reduced inclination to share the display at arm's length [73]. It also poses some of the same challenges involving effort; ubiquitous displays doubling as an information panel and an interactive surface provide minimal rest surface area as many of them are oriented perpendicular to the ground, at or above eye-level. The phenomena associated with the fatiguing nature of such mid-air or touch-based interactions directly in front of the user is termed the *gorilla-arm syndrome* [11]. Less fatiguing systems explore the tracking of finger gestures and poses for command invocation, but assume the use of supplementary finger and hand-tracking devices and cameras. Generally, gestural systems or devices are single-function or proprietary such as an armband (Myo) or a camera (Kinect, Leap Motion) as an external augmentation to the user or the environment. Finally, Social acceptability is another open issue with mid-air gestural interactions; users consider cultural appropriateness when performing gestures that are explicit and/or large in nature [76, 50].

1.2 At-Your-Side Paradigm

This thesis explores the domain of gestural input for ubiquitous computing environments. To address the aforementioned issues, we explore the space of at-your-side gestural input where the user can perform complex two-dimensional gestures on a plane parallel to the ground, with the arm resting at one's side of the body. The underlying motivations for this interaction modality include: reducing musculoskeletal fatigue involved with prolonged interactions using an arms down posture, exploring a more socially considerate and a subtle gesture space, and utilizing the capabilities provided by the smartwatch and wearable devices platform in order to minimize the use of supplementary tracking hardware and highly augmented environments. The at-your-side paradigm not only encapsulate the physical arms down posture of the interaction but also the gesture space using the smartwatch, a representation of a personalized, multi-purpose, and always-on hardware worn by the user.

Specifically, the following research questions are explored:

- How should we design gesture sets for at-one's-side gestural input? What sort of gestures make sense? What are the kinematic characteristics of side-gestures?
- Can users reproduce gestures with sufficient accuracy that the gestures can be reliably interpreted? What error rate could be expected from custom gesture sets with user-trained, user-specific gestures?
- How can we extend the vocabulary of gestures to produce large gesture sets that remain memorable and reproducible for the user? Simultaneously, what is the cost in accuracy as the space of allowable gestures gets larger?

1.3 Contributions

Overall, this thesis focuses on at-your-side gestural interaction that minimizes *consumed endurance* by allowing users to interact with their arm at rest at their side. Alongside this, we focus on user consensus both in form and performance for these gestures. We explore whether there is a consensus set of gestures or, at least, a theme of consistency among end-users. Informed by research contributions such as the Gunslinger system [45], and the Legacy Bias observation [65], we hope to further aid interaction designers in incorporating these highly subtle, low effort input gestures. We contribute:

- A gesture paradigm to address three specific issues: technical issues related to movement capture, social acceptance of gestural interactions, and user fatigue for prolonged interactions. Informed by all three constraints, we describe a design space to allow for gesture input using an arms-down posture, allowing for temporal gesture tracing on a two-dimensional plane, using on-board sensors to capture relative user arm movement. These design characteristics constitute an "at-your-side" gesture paradigm.
- A characterization of user behaviour and gesture design guidelines through a gesture elicitation study. The elicitation study shows a relatively low level of consensus in at-your-side interactions. Given the low-level of consensus, one question we had was whether or not it would be practical for at-your-side gestures to be user-specific.
- A feasibility probe into implementing a functional system within this paradigm with low training overhead to explore the effectiveness for the end-user. The larger issue involved in in this exploration is whether, with two or three templates, a user can

create a sufficiently large gesture vocabulary and then reliably expect recognition from that vocabulary. This depends specifically on the relative precision (or lack of precision) in multiple instances of any gesture, such as how much neurophysiological and sensor noise is present in any captured signal and whether that noise it too high to support reliable recognition. Using commodity hardware and a simple template-based recognition algorithm, we demonstrate the feasibility of prototype systems that can accurately recognize complex gestures. We also develop a technique for multi-character gestures that allows users to leverage n-grams of gestures to create more complex commands.

In addition, the work in this thesis answers each of the above research questions. First, considering gesture set design, our elicitation study does show that gesture sets that leverage both iconic forms and alphanumeric forms have value, in large part due to low consensus on a standard gesture set. As well, pilot studies show that simple shapes and alphanumeric gestures can be effectively recognized with over 90% accuracy with common recognizer optimization and rejection techniques. Finally, we show that complex gesture sets, dynamically created by combining one or more basic gesture templates, can also achieve the same level of 'online' accuracy with continuous, noisy data input.

1.4 Outline

The rest of the thesis is organized as follows:

- Chapter two describes the previous work in the domain of arm gesture input, focusing on various techniques for capturing arm movement, classification approaches, and challenges with gesture-based input modalities such as the *Midas touch problem* and the *Gorilla-arm syndrome* related to fatigue. We highlight existing research when it comes to the size of the gesture sets in this domain and touch on research for creating socially acceptable interactions.
- Chapter three describes an at-your-side gesture interaction paradigm, motivated by interaction design challenges associated with fatigue and social acceptance. We also discuss, in detail, how how the user can 'trace' a large variety of gestures using the arm in a low-endurance at-rest posture. We then describe a prototype system using readily-available hardware, with minimal augmentations and computationally-cheap classification techniques.

- Chapter four describes the design of an elicitation study to understand the users' behavior and preferences for gestures in this framework. We discuss the implications from the elicitation study, especially the need for diverse and personalizeable gesture sets with a preference for alphanumeric constructs that can easily be communicated.
- Chapter five describes the design and evaluation of the prototype system in a laboratory setting. We show the results of an exploratory pilot-study to perform offline recognition on the $1 and the Palm Graffiti unistroke gesture sets. We then perform a more extensive evaluation of longer, dynamically generated bigram and trigram gesture templates to stress-test the accuracy of the sytem.
- Chapter six concludes the thesis by summarizing the work and outlining the ways of extending the research in the future and areas to explore.

Chapter 2

Related Work

2.1 Overview

In this chapter, we review prior research into implementation techniques for free space gestural interfaces. We survey the breadth of implementation approaches as per the software and hardware requirements. We then discuss the real-world design tradeoffs for physiological factors such as fatigue and limits to the vocabulary and the degree of expression available to the user.

2.2 Implementation of Arm Gesture Input

In the domain of mid-air, arm-based gesture input, the system for collecting and processing input data defines its efficacy in various environments and conditions. In this section, we present the tradeoffs and challenges with building gesture interaction systems. Section 2.2.1 describes the diversity of hardware approaches to capture arm movement data that later informs the design of the software for filtering and processing. Section 2.2.2 details the challenge to extract intended data from noisy and continuous input, something common to all such systems. The rest of the section provides a brief overview of the software design for classifying gestures, detailing the requirements to provide low-latency and high precision feedback.

2.2.1 Movement Capture

2.2.1.1 Optical Systems

Computer Vision (CV) based systems, similar to human visual systems, deal with understanding and recognizing the visual structure of images and videos. Through the use of a camera, such systems are used to detect hand postures by analyzing its similarity to a set of training images. Active gesture recognition sub-fields for vision-based systems include sign language interpretation. Thang et al. [70] use camera-based position trackers to capture hand movements and generate a dataset of 95 distinct American Sign Language (ASL) signs.

Commercial products such as the Microsoft Kinect [48] and Leap Motion [40] perform three-dimensional object tracking through a set of infrared sensors. Tang [68] uses the Kinect hardware for gesture recognition by identifying the user hand in a given image, classifying the hand into a set of predefined poses, and classifying a temporal sequence of poses as a gesture.

While a plethora of vision-based hardware exists, some challenges remain with this approach. For such interactions, existing display systems need to be augmented with tracking hardware. Augmenting the environment may especially be challenging in public spaces with rigid fixtures.

Occlusion is another issue for vision-based systems. For mid-air interactions, the user arm always needs to be in a direct line-of-sight to the tracking hardware. While infrared cameras can help with tracking objects in various lighting conditions, the camera field-of-view limits tracking out-of-sight and multiple of objects. Liu et al. [45] have tried to minimize the line-of-sight and occlusion issues by strapping the tracking system directly on the user.

2.2.1.2 Non-optical and Inertial Systems

While approaches to measuring musculoskeletal activity for motion capture vary depending on the sensors and instrumentation available at one's disposal, the two primary techniques include electromyography (EMG), and through the usage of inertial measurement units (IMUs) due to its availability in commodity hardware.

2.2.1.2.1 Electromyography (EMG)

Figure 2.1: Myo arm band recognizes five basic distinct gestures [67].

Electromyography (EMG) has recently gained prominence in the domain of gesture recognition. This technique is used to record musculoskeletal activity by measuring the electrical response produced by the skeletal muscles. Consumer products such as the Myo armband device [69] use EMG sensors, combined with a gyroscope, accelerometer, and magnetometer, as a mechanism for sensing electrical activity in the forearm sensors to classify a set of five gestures. Figure 2.1 shows the gestures detected by the Myo armband.

Xiong et al. [81] use postures detected by EMG sensors to perform mouse actions and use the inertial measurement sensors to perform mouse navigation. Similarly, Haque et al. propose a pointing technique using the Myo armband, Myopoint, as an on-body alternative to camera-based approaches [26]. This technique uses the coarse arm movements for pointer navigation and fine finger movements for pointer actions such as clicks.

EMG techniques eliminate environment augmentation and avoid the problem of occlusion and lighting in vision-based systems. However, EMG sensors are far from a commodity; the users need to purchase specialized EMG devices such as the Myo armband and equip them as needed given the limited battery life in both active and standby mode.

2.2.1.2.2 Intertial Measurement Unit (IMU)

The rapidly growing ubiquity of smartphones, smartwatches, and activity trackers has had a direct implication in the gesture recognition domain. Such devices come equipped with a variety of motion sensors, an accelerometer and a gyroscope at the very least, comprising the inertial measurement unit (IMU). The IMU is used to detect the relative inertial movement and absolute orientation changes in the device. While the primary purpose for the IMU in such devices is for recognizing various motion-based activities

such as walking, running, and exercising, recent work has explored in using the IMU for mid-air gestures. Watchpoint by Katsuragawa et al. [31] and Smartcasting by Pietroszek et al. [58] are examples of using the hardware IMU for exploration of the smartwatch and the smartphone devices for distant pointing.

Porzi et al. use the IMU in the smartwatch for assisting persons with visual impairments [60]. They evaluate the performance of the system through a set of eight abstract-shape, two-dimensional gestures performed by user's arm movement to demonstrate its viability.

Commodity hardware such as the smartwatch can be altered in software and hardware to enhance sensing capabilities. Laput et al. stretch the built-in accelerometer by increasing its sampling rate to 4000 Hz to analyze micro-vibrational patterns for bio-acoustic sensing [39]. Additional sensor systems can be added on or around the device for detecting microgestures and finger postures [82, 13].

The widespread availability of the IMU in commodity smartwatch hardware makes it an opportunistic form factor to sense gestural input. The "always on" and "always available" capabilities further makes the watch a convenient and inexpensive platform. Research has leveraged these advantages through various interaction paradigms [14, 31] and developed toolkits [28] for the ubiquity of smart embedded devices.

2.2.2 Segmentation of Gestures and the Midas Touch Effect

A major factor in the adoption of motion gestures is its usability in real-world applications: a gesture system should be optimized to be resource- and power-conscious, and its ability to filter unintended motions and noise to acquire clean, segmented gesture data. In interaction design, this challenge is further exacerbated by the dichotomy between the interpreting the user's arm movement as an input modality or to navigate in the user interface, termed the *Midas touch effect* [5].

A gesture delimiter helps in both such cases where a user can perform a pre-defined delimiter, in the form of a gesture or other means to trigger or activate a resource-intensive gesture recognition session. Within the session, a delimiter can also segment between two distinct gesture tasks. Figure 2.2 illustrates this process of initiating gesture sessions and segmenting gestures.

The implementation of a delimiter can vary in nature and comes with different tradeoffs. An explicit delimiter, such as a pre-defined motion gesture, can increase the confidence of a gesture system in identifying clean gesture segments. Ruiz and Li [63] propose a "double flip" input delimiter for smartphones as being extremely resistant to

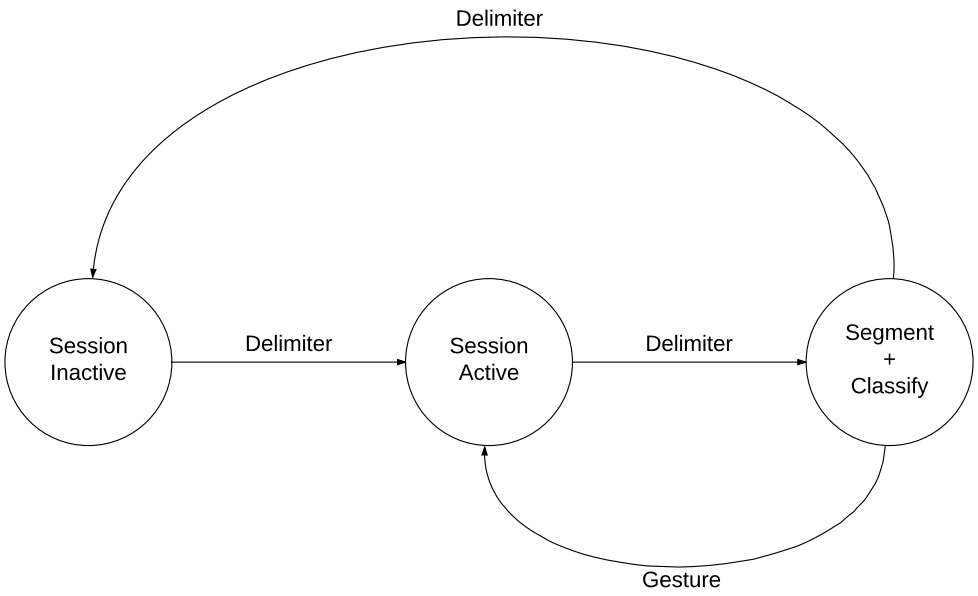

Figure 2.2: State machine for gesture session activation and segmentation within gestures.

false positives. The tradeoff for such explicit gesture delimiters are in degraded user experience: the system expects that the user must remember to perform a pre-defined gesture to segment input. Motion delimiters can also be difficult to generalize across a variety of users with different physiological characteristics and handedness.

Automatic segmentation can eliminate the need for explicit motion delimiters by putting the onus on the system to dynamically segment gestures. Kratz and Back explore, through the use of machine learning based classification, detecting distinct phases of gesture input [37]. Such classification approaches are not without drawbacks: generating and annotating the initial training dataset can be time-consuming, and the model has to adapt to ever-evolving user kinematics.

At the intersection of explicit and automatic approaches to segmenting everyday movement from input gestures, researchers have also explored temporal delimitation. Temporal delimitation can be used as a variation for implicit gesture delimiters: device displacement within a threshold in a fixed time period can infer an input segment. Natural

language processing [20] and vision-based systems [80] use temporal thresholding as part of the segmentation heuristics.

2.2.3 Recognition Strategies

Arm gestures, commonly used as a mode of communication with other humans, have naturally been adapted as a form of human-computer interaction. Yet, some limitations remain in terms of sensing and categorizing gestures due to technological limitations. Once a gesture have been sensed using equipment such as depth cameras or IMU devices, a mathematical representation is used to describe its distinguishing features such as joint locations, motion velocity, and shape parameters. The spatio-temporal nature of arm gestures make it crucial for classification algorithms to account for both spatial and temporal features in order to provide robust recognition. The high-level primary approaches used for gesture classification can be broadly categorized into *template-based classifiers* and *machine learning-based classifiers*.

2.2.3.1 Template Recognizers

The most significant challenge with motion gesture input, regardless of the sensing technology used, is recognizer reliability [42, 54]. To address the issues of recognizer reliability, many gestural input systems use relatively small gesture input languages [55, 64, 78]. As an example, In studying reliability of recognition, Negulescu uses 4 input gestures [55]; in their consensus set of motion gestures Ruiz et al. depict 12 gestures (mapping onto 14 commands) [64]; and a frequent benchmark surface gesture set for recognizer prototyping, the $1 gesture set, has 16 gestures [79]. Various technologies have also been exploited to recognize more subtle wrist and finger gestures [19, 33, 34, 38, 47], including IMU sensing [74], but, again, with dictionary sizes between four and fifteen gestures. In practice, the human interface guidelines provided by Apple for smartwatch applications design explicitly recommend against using IMU-sensor data as an input modality due to its imprecise and noisy nature [6].

In template recognizers, the input data is matched against a set of reference templates in order to find the “closest match”. Template gestures first need to be generated either algorithmically or from the user to be matched with the data. A gesture template can be comprised of a multi-dimensional temporal signal. Once a finite set of templates are collected or generated, the test data sequence is compared with the entire template set. In terms of a threshold, the input is either classified or rejected depending on its structural

similarity with the most similar template. The larger the template set, the longer it will take for the system to classify input. Techniques such as *Dynamic Time Warping (DTW)* are used to template-match temporal sequences and more recently, gesture motion sequences [16]. DTW, commonly used in speech processing to compensate for varying speeds in speaking [66], can be conceptually applied to match motion gestures by compensating for different speeds of arm motion. This type of a bottom-up pattern matching approach is better suited to dealing with raw motion data in a limited dataset in contrast to abstract geometric models extracted through machine learning techniques described in the next section.

2.2.3.2 Learning-based Classifiers

To enhance classification accuracy in realtime conditions, classifiers are trained on a dataset of gestures in order to extract the "best" features [18]. Learning algorithms such as hidden markov models (HMMs) and AdaBoost can adaptively extract and concatenate dataset features to strengthen the classifier in a given dataset. Such techniques have been explored for hand and pose detection in real-world settings [61]. The training process for these learning algorithms generally require a large training dataset and computational power, often times performed in application-specific integrated circuits (ASICs).

2.2.4 Recognition Challenges – Balancing Precision and Recall

Even in the most optimial settings, there is uncertainty and noise inherent in any gesture recognition task. As a result, errors and low-confidence classifications are unavoidable. The rejection mechanism, thus, acts as a way to reject the classification instead of providing a false classification. Previous work by Chow explores the tradeoff inherent in rejecting low-cofidence classifications, the reject rate, and finding a threshold that can minimize the error rate [15]. Deriving an optimal rejection scheme is a known challenge in gesture and handwriting classification [25]; the fault-tolerance of different situations and environments can drive the error-reject tradeoff.

2.3 Usability Constraints on Gesture Input

Acceptance is the key usability aspect for designing gestural interfaces involving explicit arm motion. The concept of social acceptance when designing gesture interfaces has

not been explictly defined or quantified, in large part due to its intangible nature. The next section details of social and cultural acceptance of gestural interfaces, especially in public and semi-public contexts, to inform design considerations. The following chapter discusses attempts to quantify physiological acceptance of gestural interfaces and ways to measure fatigue.

2.3.1 Social Constraints on Arm Gestures

Environment and context highly shape the way people use gesture-based systems. Williamson and Vennelakanti show the diversity and similarities in the selection of gestures and the methods to perform them in different cultural settings [76]. Their work indicates that the size and shape of gestures are important factors to consider when designing gestural interactions to be used in social spaces.

In the same vein, Montero et al. present other factors that influence the social acceptance of gestures that include "culture, time, interaction type and the users position on the innovation adoption curve." [50]. This work puts forth the notion that the user's perceived interpretation of the gesture to a bystander is a crucial factor in determining the gesture's social acceptability.

Besides social perception of explicit mid-air gestures, the perceived physical space available for interaction also changes the way people interact with systems in public spaces. Azad et al. [7] explore the impact of social dynamics on display usage characteristics such as positioning and orientation. Wallace et al. [73] explore the concept of a "personal space bubble" that the users conceptualize as a personal space boundary when co-operatively interacting with large public displays. Their work argues for consideration of personal space as a design constraint with relation to the size of the screen when designing interactions, especially with direct touch input.

2.3.2 Physiological Factors

Free space gestural input is an effortful form of communication, giving rise to terminology such as the well-known *gorilla-arm effect*[11, 27] and metrics such as *consumed endurance* [27], which characterize the effort one must expend to support interaction. This metric provides a method to quantify arm fatigue via Kinect motion capture and has been used to evaluate less fatiguing freehand gestures [45]. A key result of the evaluation by Hincapie-Ramos et al. is that interactions requiring lower shoulder torque will be less fatiguing. The

implications of arm fatigue are wide ranging from general discomfort when interacting with systems, to systems going unused. Hincapie-Ramos et al. define *CE* as follows:

$$CE(T, TotalTime) = \frac{TotalTime}{E(T)} * 100$$

where

- CE is the ratio of the interaction time and the computed endurance time
- E(T) is the continuous time, in seconds, during which a user can interact before resting their arm

$E(T)$, calculated as follows:

$$E(T_{shoulder}) = \frac{1236.5}{(\frac{T_{shoulder}}{T_{max}} * 100 - 15)^{0.618}} - 72.5$$

This equation is asymptotic at 15% of the maximum shoulder torque, meaning that any interactions involving less than 15% of the shoulder torque can be sustained for indefinitely long periods of time. The definition of torque, as $\vec{T} = \vec{F} d \sin\theta$ suggests that the torque applied on the shoulder equals zero, to a close approximation. As a result, $E(T_{shoulder})$ falls safely below the 15% asymptote. As a natural corollary, for subtle arm movements on the side, the resulting CE value for such at-your-side interactions are undefined and possibly sustainable for very longer periods of time, indicating further exploration to validate and update existing metrics for this posture. For example, in a recent study of freehand interaction in the living room, 68% of gestures were changed over time, with researchers motivating these changes due to users experiencing fatigue during interaction [41]. Researchers have also explored the use of finger-based gestures for command invocation [13]. Such interaction techniques are potentially less fatiguing but assume the use of supplementary finger and hand-tracking devices and cameras [45].

Recent work by Jang et al. [29] propose a fatigue estimation model using a vision-based skeleton tracking system and estimating more subjective fatigue measures (e.g. the Borg scale [10]).

Alongside effort, free space communication is also noisy; it is hard to exactly replicate motions in free space unless significant constraints are placed on the form of input path. Finally, sensing technology is challenging. Large-scale gestures are easy to sense from afar, but small-scale and subtle movements require spatial, set size, or kinematic constraints.

2.4 Gesture Set Design

Gesture-based input modalities involve designing motion gestures that are easy to perform, intuitive, and serve as direct metaphors for task at hand. Designers of the gesture sets must consider the many factors when choosing appropriate gestures ranging from its naturalness to the ability for the classifier to accurately classify it. Gesture elicitation studies have widely been used to inform the design of the gesture sets.

2.4.1 Elicitation Studies

One valuable tool for designers and researchers seeking to understand "naturalness", "mental models", or "consensus" in terms of input (or cause) and invoked computational behavior (or effect) is the elicitation study. Elicitation studies have been commonly used in gestural input space to inform the design of gesture sets [79, 65, 64]. While the gesture sets generated by elicitation studies are not necessarily the definitive gesture set for use in an interaction paradigm, their value lies in adding to the understanding of the designer. The designers role is to resolve real world constraints, including data from the elicitation study, data on recognizer reliability, sensing capabilities, etc. to arrive at a gesture set that effectively supports input.

A common rate of evaluation for such elicitation study is with agreement rate. An agreement rate, AR is computed for each task as a single value between $[0, 1]$ and corresponds to the degree of consensus among all participants. Previous gesture elicitation studies [79, 65, 64] used agreement score as initially calculated by by Wobbrock et al. [79], which did not accurately de-emphasize gestures with zero agreement. Recent work (e.g., [13]) has employed an updated agreement rate formula proposed by Vatavu and Wobbrock [71]:

$$AR(r) = \frac{|P|}{|P| - 1} \sum_{P_i \subseteq P} \left(\left| \frac{P_i}{P} \right| \right)^2 - \frac{1}{|P| - 1}$$

Where P is the set of participants sampled, and P_i represents a set of agreed upon gestures. Importantly, Vatavu et al. [71] also provide guidance for interpreting AR, with suggested levels of *low* ($\leq .1$), *medium* ($.1 - .3$), *high* ($.3 - .5$), and *very high* ($\geq .5$) agreement; these bins were also used as a basis for classifying the elicited data in this study.

Device	Dictionary Size 1s	10s	100s – 1000s
Smartwatch	[6, 35, 60]	[17, 82]	[49]
Smartphone	[30, 36, 54]	[43, 63, 78]	[1]
Other	[9, 26, 44, 45, 72]	[2, 12, 23, 51, 65, 77]	[24]

Table 2.1: Related research categorized by device and dictionary size.

In practice, few gestures may have high agreement but nonetheless provide valuable insights into how the users think in given constraints and the characteristics of the gestures they prefer. Research has shown such gestures to be more natural [52] and easy to recall in multiple domains including mobile devices [64] and multimodal systems [51].

2.4.2 Challenges in Gesture Set Design

Creating gesture sets involves a two key considerations for the designers: transference – gesture sets heavily influenced from other interaction domains, which may be applicable to varying degrees; and long-term usage – the correlation between gesture sets and user preferences to use them in public and social contexts.

Recent work identifies two aspects of interaction to address some of the challenges associated with free space gestural input. Recent work by Ruiz and Vogel on *Legacy Bias* [65] explore the issue of fatigue when using gestures to control an external display. They show, through an elicitation study, that attaching simple weights onto the users' arms can prime them to consider fatigue-related issues when eliciting gestures. The second piece of work that significantly influences us is work on the use of wearable devices to support gestural sensing. These wearables take several forms – body mounted Leap Motion devices [45], smartwatches [17, 31, 49, 82], special purpose input devices [26] and permit gestural interaction with nearby computation via sensing technologies.

2.5 Summarizing Arm Gesture Input Research

The focus of this thesis is arm gesture input, primarily in both the arms-in-front and arms-down posture. This chapter explored related work in this domain, from approaches to capture movement data to using different techniques to classify input. Different selections in interaction design of gesture techniques comes with tradeoffs for the system: explicit and coarse-grained input can be easy to recognize but fatiguing to perform while subtle input can be more socially-inconspicuous but a challenge to capture.

Table 2.1 summarizes recent work in free-space gesture recognition, with a breakdown of supported devices and dictionary sizes. When moving towards more general support for the myriad devices available in ubiquitous or multi-device settings, it is difficult to determine how large of a gestural language is practical. Are four to fifteen gestures sufficient for real-world contexts? Given the ubiquity and variety of smart and embedded devices [21], we suspect not. This, then, begs the question of how potential users might handle larger gesture sets and whether recognizers of free-space gesture sets can function given neurophysiological noise in input and noisy sensor data.

These challenges persist in any gesture paradigm involving arm-motion input. In the following chapters, we detail an interaction paradigm designed around these systematic and physiological challenges and describe the tradeoffs made in the process.

Chapter 3

At-Your-Side Gesture Paradigm

3.1 Overview

The at-your-side interaction paradigm describes an arms-down, always-available, socially conscious, gesture drawing technique using commodity hardware and sensors. In the previous section, we described some of the challenges and tradeoffs with different approaches in designing gesture interactions. Considering these tradeoffs, we formed three primary design principles to prioritize real-world usability in gestural interfaces:

- *Minimize fatigue* - The system should try to minimize the amount of endurance placed on the user
- *Always available* - The system should use readily hardware readily available or within reach and avoid additional setup of sensors and cameras
- *Rich and socially acceptable* - The system should support a large gestural vocabulary that is sufficiently expressive and yet accommodating to public and social contexts through subtlety

These design constraints drive the basis of at-your-side gesture input accounting for real-world usability and user experience factors. the *Always available* criterion is satisfied by using the built-in IMU from an Android smartwatch, *LG G Watch R*, as a prototyping device. The smartwatch platform, in this case, provides an opportunity to leverage the fact that it is always worn by the user and conveniently houses positional sensors. In turn,

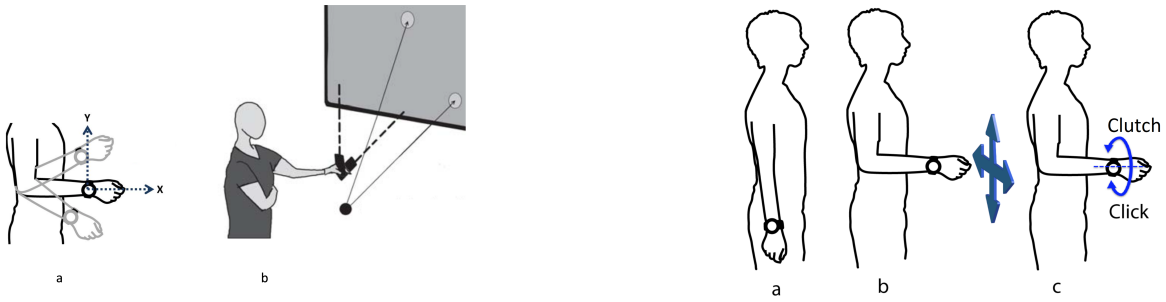

Figure 3.1: Left: Pointer manipulation in the rotation techniques, W1RR and P1RR, is governed by the orientation changes of the control device. (a) For W1RR, the user must sweep the entire forearm to cause changes orientation of the smartwatch. (b) For P1RR, the user can use the wrist sweeps as a more subtle form of manipulation; Right: W1RR can calibrate the center frame of reference from inactive state (a) to raising the arm in front (b) to switch to manipulation mode. From this mode, a 45 wrist flick outwards and back can be used to trigger a click action (c).

the IMU within the smartwatch can be sampled for relative positional measurement of the user's arm and can be used to draw gesture traces, much like handwriting, to provide subtle, implicit input.

3.2 Formative Observations on Side-Gestures

Our work is motivated by the exploration into using the smart devices, without augmentations, as a viable pointing device with Watchpoint by Katsuragawa et al. [31] and Smartcasting by Pietroszek et al. [58]. The systems leverage the built-in IMU for the cursor navigation and action tasks using mid-air arm rotation, as shown in Figure 3.1.

An arms-down posture was previously explored by Gunslinger [45] as a form of input minimizes the torque applied to the shoulder, the primary source of user fatigue quantified with *Consumed Endurance* [27]. In our research group, we have noted that problems of fatigue during gesture input significant influence how users perform gestures. While users may initially perform large gestures 3.2 (Left), over time we see them reduce the size and force to limit fatigue 3.2 (Right). We have coined the term 'the wiimote effect' to describe this phenomenon. This term is based on the similarity of user behaviour to behaviours when using the Nintendo Wiimote. In particular, when users are beginning to use the Wiimote for console games such as tennis, golf, and bowling, they perform large gestures

Figure 3.2: *The Wiimote effect*: Fatigue causes the users to slowly retract the arm closer to the body over time.

similar to real-life gestures in these sports. However, as they become proficient, they swing less, realizing that a simple wrist flick can minimize fatigue and still provide exactly the same effect – sometimes even an enhanced effect – with much less movement.

3.3 A Prototype System for At-Your-Side Gesture Input

Before embarking on a deep exploration of the at-your-side gesture input, we designeda prototypesystem for use during design exploration. The experimental apparatus consisted of an LG G Watch R running Android Wear platform version 1.3, a Bluetooth-connected LG Nexus 5 smartphone with Android version 6.0.1, and a PC connected via TCP/IP. Sensor data from the smartwatch was time-stamped and serialized into a JSON array before being transferred to the recognizer application. Additionally, the converted two-dimensional drawing plane data was recorded for each trial as a continuous stream including noise and non-relevant motion. The recognizer then annotated each sample with the user's detection

state, the recognized gesture, test instance information, and whether the prediction was successful or unsuccessful. This flow of data and its mapping at various points of the communication pipeline is illustrated in Figure 3.4.

3.3.1 Mapping Sensor Input to a Two-Dimensional Trace

We map the changes in device orientation (Δ_ω) on the Yaw and Pitch axis (in radians) to displacements of the cursor d, using the following function:

$$d = \Delta_\omega \times G(v)$$

$G(v)$ corresponds to the value returned by a piecewise linear CD gain function of the form:

$$G(v) = \min\left(\max\left(s \times v + i, G_{min}\right), G_{max}\right)$$

with v the five-sample average velocity change in orientation in radians per second. The parameters G_{min}, G_{max}, i, and s are defined empirically for the range of sensor values produced by the gravity sensor on Android. Changes in orientation around the Yaw (respectively Pitch) axis are mapped to x (respectively y) displacements of the cursor. This technique, in concept, is similar to Tiltcasting by Pietroszek et al. [59] but adapted to draw two-dimensional gestural traces in an arms-down posture. The sensitivity is limited to movements of the whole forearm as the wrist tilt is naturally not sufficient to cause significant changes in the orientation of the watch. Figure 3.3 illustrates the arms-down posture with a two dimensional tracing plane. The data is transferred over Bluetooth to the companion phone using the platform messaging API. The data is then logged and off-loaded to a PC for further filtering and recognition.

3.3.2 Recognition Strategy: Dynamic Time Warping

We used the Dynamic Time Warping (DTW) algorithm, a simple template-based pattern matching technique widely used for speech recognition systems and other continuous data streams [8]. Typically, DTW measures the similarity index between two temporal signals that can vary in time or speed and computes the warping distance to match the two signals by either stretching or compressing the time axis. The classifier then computes the relative similarity of the sample with all the templates in the model and makes a prediction based on the lowest DTW warp distance, a classification technique known as 'nearest-neighbour classification'.

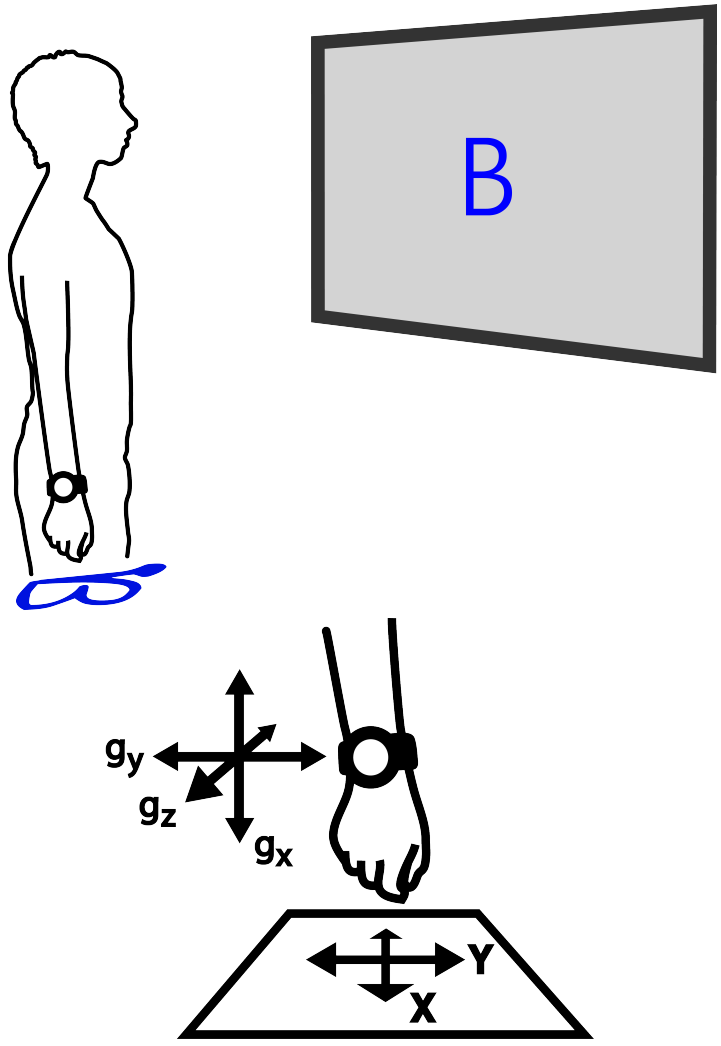

Figure 3.3: The interaction design uses an arms-down posture to use draw two-dimensional gesture traces at the side of the user. Gesture traces are created by detecting the relative arm displacement in an imaginary bounding box to the side of the user.

We applied normalize the gesture size and position by uniformly scaling the bounding box of the trace on the drawing plane. DTW was applied to the normalized sample traces and normalized template traces. The classification was performed on clean, segmented data in an offline setting.

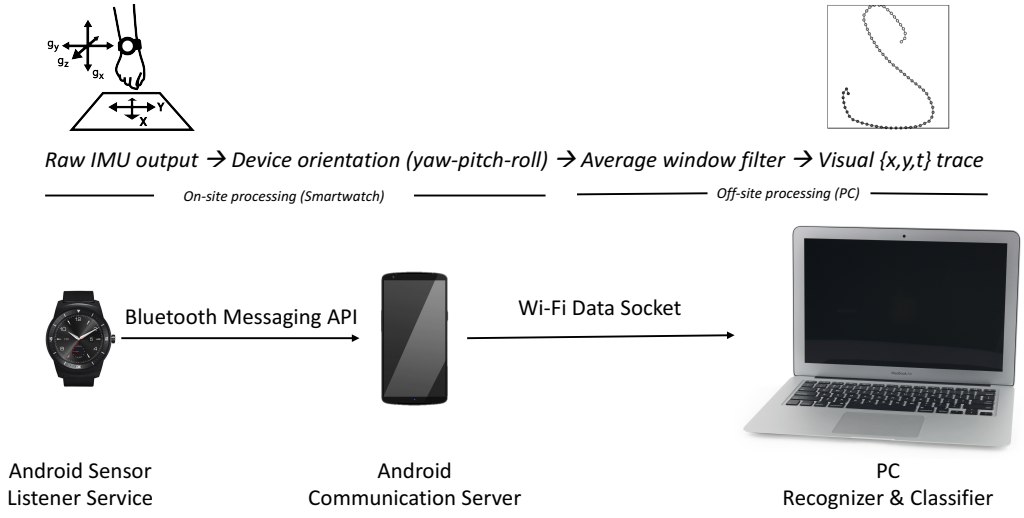

Figure 3.4: IMU data is sampled from the smartwatch and converted to relative device orientation. This data is then packed and sent to the off-site PC for classification.

3.4 Pilot Study

To explore the potential of the smartwatch as a platform to detect motion gestures we conducted a pilot study. One requirement that we noted in our elicitation study was both ideographic and alphanumeric data were important to assess as input. We could, conceivably, have allowed users to define their own gesture sets, but that would tell us little about feasi- bility, as distinctiveness of gesture sets designed by different users would impact feasibility.

To control for distinctiveness variation in gesture sets and to look more carefully at reliability and training of single gesture input, we used the $1 gesture set [78] and Palm Graffiti [23]. As a starting point for simple recognition, we use the $1 gesture set because many researchers have assessed recognition accuracy against it and it represents an ideographic gesture set of approximately the correct size. We use Palm Graffiti because it is another instance of a unistroke gesture language, it is 60% larger than the $1 gesture set (26 vs. 16 symbols), and its alphabetic correspondence means that if one wishes

to extend the language, combining gestures into bigrams and trigrams has an intuitive mapping onto linguistic constructs. Figure 5.1 illustrates the unistroke gesture sets used.

Participants performed an off-line recognition task without any visual feedback. Linear acceleration, gyroscope, and orientation and rotation vectors were logged on a LG G Watch R, running Android Wear at 200 Hz. We then examined the input stream and ensured good segmentation of our input data. Finally, we ran the gestures through our recognizer to assess accuracy readings for each gesture set. We did not optimize the recognizer; our goal was simply to test the potential of recognition and the need for training data. In total, 16 gestures $\times$ 10 repetitions $\times$ 8 participants = 1280 gestures were collected for the $1 Gesture set and 26 $\times$ 10 $\times$ 8 = 2,080 gestures for Palm Graffiti.

3.4.1 Results

Recognizer accuracy using leave-n-out evaluation. Overall, the prototype achieved a recognition rate of 94% ($\sigma = 4.69$) for $1 input and 89% ($\sigma = 4.78$) for Palm Graffiti input with maximum recognition occurring using 9 templates for each gesture (leave-one-out). The recognition rate for the $1 Gesture set and Palm Grafitti with one template per gesture was 88.05% ($\sigma = 5.3$) and 82.3% ($\sigma = 6.2\%$) respectively (Figure 3.5). Recognition rate increases with the number of templates, but stabilizes with three templates.

Overall, while surface gesture error rates using prototyping recognizers are typically above 95% [78]), a recognition rate in excess of 90% for an unoptimized recognizer using only three training samples is promising. There are many ways that gesture recognizer behavior can be optimized, including using n-best lists and confidence thresholds and reject decisions [15].

3.5 Summary

This chapter described the at-your-side gesture paradigm, an effort-minimizing gestural technique grounded with the *Consumed Endurance (CE)* metric to prolong mid-air interactions. A natural corollary of the CE calculation is that, to prevent fatigue, gestures can take place at a user's side instead of in the air in front of them. For such movements, the CE value is undefined when the arm is at-rest position due to a negligible amount of shoulder torque, meaning the interaction can theoretically be performed for long periods of time [27].

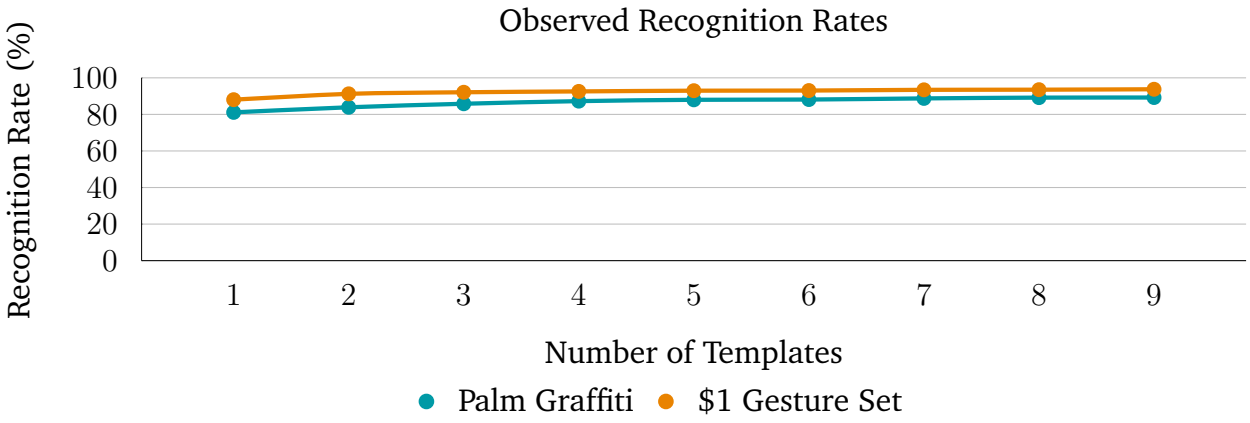

Figure 3.5: DTW recognition rates with a user-dependent template for the $1 Gesture set and Palm Graffiti, by number of templates per gesture.

We then presented a feasible prototype system as a feasibility measure adhering to these design constraints by converting user arm movement to two-dimensional gesture traces. Based on the pilot study to classify the $1 and the Palm Graffiti gesture set, the system shows promise in accuracy using a relatively naïve implementation of DTW. Our data indicate, first, that even with three templates, a user can expect to receive reasonably high accuracy with at-the-side gestures. Furthermore, in analyzing data from our recognizer, one of the attributes of the data that we noticed is that positional data, particularly as the offset from gravity was most revealing in our input. Essentially, because the participant's arm was hanging at rest at his or her side, we could view side-gesture input as a two-dimensional drawing on a plane, and we leverage this in the recognizer design in the next study we describe.

Overall, while surface gesture error rates using prototyping recognizers are typically above 95% [78]), a recognition rate of more than 90% for an unoptimized recognizer using only three training samples can be further explored. There are many ways that gesture recognizer behavior can be optimized, including using n-best lists and confidence thresholds and reject decisions [15].

Chapter 4

Elicitation

4.1 Overview

In this chapter, we describe a gesture elicitation study to understand the types of gestures preferred by users. The study was conducted in order to understand the mental model of the users interacting within the at-your-side interaction paradigm. The observations corresponding to the nature of the gestures the participants create directly informed the design space of the system.

We followed the protocol used in previous elicitation studies for free space gestures [79, 65, 64], where participants were invited into a laboratory and asked to design and enact gestures for a given task. The participant is then engaged in a participatory design discussion involving their design process and evaluation of the design in terms of social and physiological factors. The study provided an opportunity to elicit a gesture language appropriate for the subtle, at-your-side motions we wished to develop. We analyzed proposed gestures for general trends among participants: gesture length, type, complexity, behaviour for forming gesture chains or sequences, and recall rate.

4.2 Task Taxonomy

Our scenarios also included commonly used semi-public applications like Google Maps, Netflix, and Spotify (Figure 4.1). We mimic the set of action metaphors for a new set of tasks in the context of familiar public displays: an airport departures screen, and a

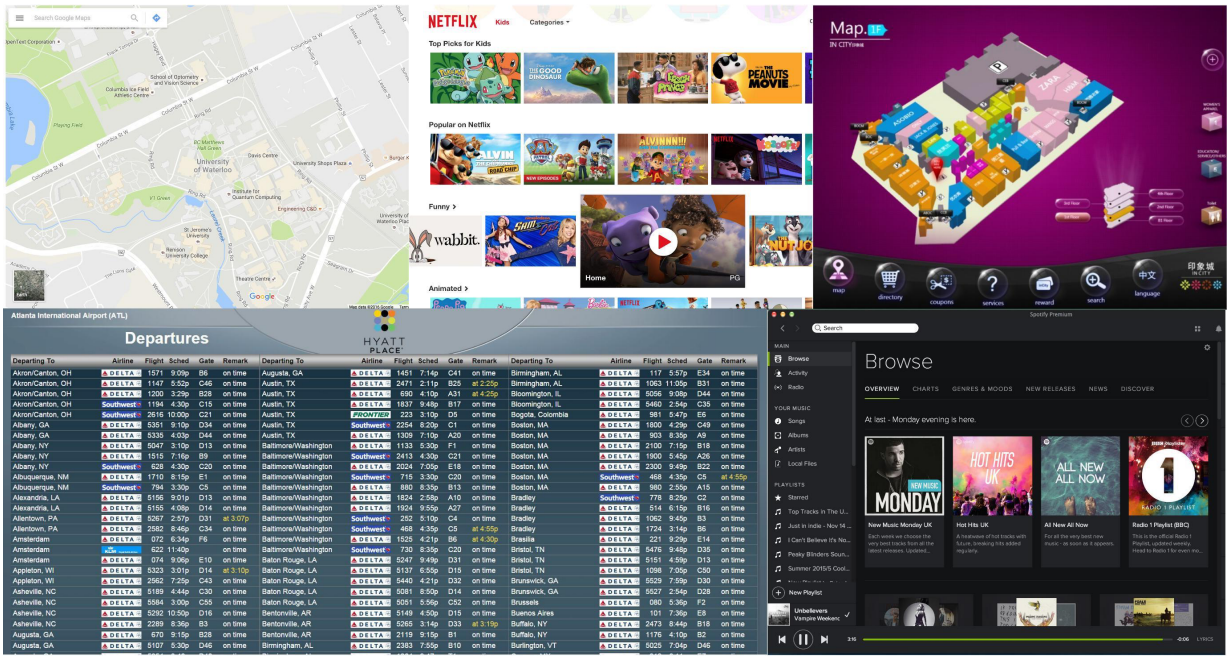

Figure 4.1: Interface mock-ups as displayed on a large projector for participants to imagine the task and elicit a gesture. From left to right: Google Maps, Netflix, Mall Display, Airport Departures, Spotify.

shopping mall kiosk. These contexts are designed to be a mix of things one might interact with in a public setting such as flight departures and mall directory navigation. More private contexts such as controlling the Netflix application or Spotify represent interfaces primarily used at home. The Maps context is a canonical application that applies well across all both social and private settings.

Our study uses a set of tasks (Table 4.1) inspired from previous research in elicitation tasks for mobile and whole-body interactions [65, 64]. Ruiz et al. [64] use ‘action’ and ‘navigation’ tasks as the high-level taxonomy for eliciting gestures. Recognizing that public display interfaces are inherently more dense with information in tabular or graphical layouts due to the real-estate available, we include an additional category of ‘filter’ tasks to encapsulate information selection in tabular, graphical, and list formats. Emphasis was placed on a fair representation of tasks across contexts and categories instead of exhausting the feature set of a specific application.

	Type		
Context	**Navigation**	**Action**	**Filter**
Google Maps	Zoom In/Out Pan Up/Down Pan Left/Right	Toggle View Show Menu Activate Search	Reset Location
Netflix	Select Next/Previous Page Left/Right	Play Item Activate Search Exit to Home Screen	Jump to Movie
Mall Display	Floor Up/Down		Jump to Store Jump to Floor Level
Airport Departures			Jump to Flight Filter Delayed Flights Clear Filter
Spotify	Next/Previous Song	Play Item Pause Item Replay Item Volume Up/Down Favorite Item	Jump to Menu Item

Table 4.1: The 34 tasks presented to participants during the elicitation study. Tasks were categorized by type: Navigation, Action, or Filter.

4.3 Experiment Design

Images for each of the application interface were displayed on a 60" rear-projection system for simulating large display interaction (Figure 4.1). The interface mock-up for the interactive mall display application was retrieved from the personal website of a designer, Ge Zhang[1]. All the participants were required to wear a LG Watch R to simulate a working system. As the study was primarily constructed to be a participatory design excercise focused on high-level discussions, raw sensor data was not recorded for the elicitations.

Twenty paid participants (10 female) between the ages of 21 and 55 ($\mu = 26.6$, $\sigma =$

[1] gezhang.com

Figure 4.2: Participants in the elicitation study designing gestures for various tasks using a think-aloud approach.

7.6) were recruited. Participants were asked to use their dominant hand when eliciting gestures for each task, with three of the participants being left-handed. Every participant received $10 remuneration. Participants were mainly university students in a Science, Technology, Engineering, or Mathematics field. We did not require or control for prior experience using or developing gestural systems.

At the beginning of each experiment, we explained the study to the participant and asked them to wear a smartwatch and imagine interacting with a large display in a public setting (Figure 4.2).

We elicited gesture preference for the tasks (Table 4.1) using a think-aloud approach. For each task, participants elicited one gesture and performed it 4-5 times before finalizing their decision. Participants were instructed to choose any shape, symbol, movement or a combination of shapes, symbols, or movements for each task. The tasks were grouped by application and were presented one-by-one. We did provide some flexibility in ordering; for example, for tasks such as *Pan Up*, participants were free to clarify and choose a set of gestures for panning in the other three directions: *Pan Down*, *Pan Left*, and *Pan Right* at the same time.

Throughout the study, we did not want the participants to think of the capability of the gesture recognizers, and so we did not provide any feedback or confirmation to participants to remove the *gulf of execution* [56]. Instead, participants were instructed to only concern themselves with designing what they considered to be the most 'natural' interaction, assuming no limitations of such a gesture recognition system.

Once participants had selected a gesture, participants rated the gesture for its level of fatigue, whether it was a good match for the task, and whether they felt comfortable performing the gesture in public. These assessments were collected using a seven-point Likert-scale.

Finally, at the end of the study, a recall test was performed by having the participants try to enumerate the elicited gesture for each task in a randomized order. As a follow-up, participants were asked to comment on general design issues for free space interaction,

such as “What constitutes as good gesture for a given task?”, “What are some other applications or use cases for a gesture-based input system?”, “Do you prefer alphanumeric gestures or ideographic gestures and why?”. Participants’ responses were later manually transcribed by the researcher. The video of the study was then transcribed to further identify common themes and trends using inductive, bottom-up reasoning. Each session took approximately one hour to complete.

4.3.1 Dataset

We collected, across 20 participants and 34 tasks, we collected a total of 680 gesture samples and corresponding recall information as transcribed from the design interview. We collected a total of 2,040 Likert-rating self-evaluation used for engaging the participants in a design discussion about the social and physical challenges involved with the at-your-side paradigm.

4.3.2 Diversity

We first grouped together gestures that were *visually identical* — gestures described by participants with the same visual trace. For example, a ‘straight vertical line’ gesture trace would be identical when drawn either up to down or down to up. The gesture elicitation traces were roughly estimated from the design interview transcription and experimenter logs. However, it would be different from a ‘horizontal line’ because the end trace of the two gestures visibly differ if drawn on a piece of paper, as illustrated in figure 4.3. The stroke-order is thus not considered to be the differentiating factor if the end visual trace of the gestures is identical. This constraint is strictly used for estimating the diversity of the gesture vocabulary; the participants were still able to use the same gesture in a different stroke order for different tasks.

4.3.3 Consensus

From the gestures collected from participants, we grouped together identical gestures to evaluate the degree of consensus. Using this reasoning, we evaluated the degree of consensus among all the participants by using Vatavu et al.’s methodology [71]. An agreement rate, AR is computed for each task as a single value between $[0, 1]$ where a high value means that many participants chose the same gesture, and a low value

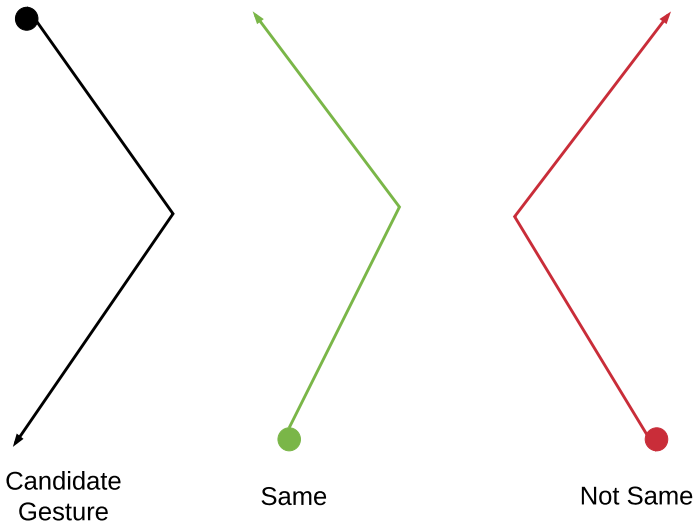

Figure 4.3: A visual trace of the gesture is used for classification: gestures are classified as identical if their visual trace is identical, regardless of the stroke-order.

means fewer participants chose the same gesture, hence more diversity in the gesture. An agreement rate, AR is computed for each task as a single value between $[0, 1]$ and corresponds to the degree of consensus among all participants. Previous gesture elicitation studies [63, 65, 79] used agreement score as calculated by by Wobbrock et al. [79], which did not accurately de-emphasize gestures with zero agreement. Recent work (e.g., [13]) has employed an updated agreement rate formula proposed by Vatavu and Wobbrock [71]:

$$AR(r) = \frac{|P|}{|P| - 1} \sum_{P_i \subseteq P} \left(\left| \frac{P_i}{P} \right| \right)^2 - \frac{1}{|P| - 1}$$

Where P is the set of participants sampled, and P_i represents a set of agreed upon gestures. Importantly, Vatavu and Wobbrock [71] also provide guidance for interpreting AR, with suggested levels of *low* ($\leq .1$), *medium* ($.1 - .3$), *high* ($.3 - .5$), and *very high* ($\geq .5$) agreement. AR for all tasks included in our study are shown in Figure 4.4.

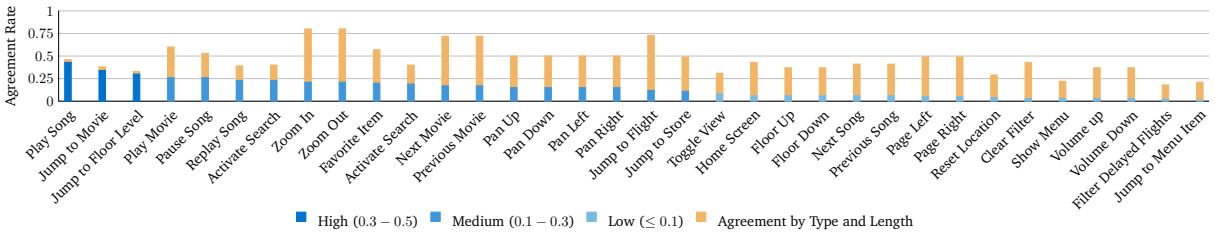

Figure 4.4: Agreement Rates (AR) for suggested gestures in terms of its visual trace, by each of the 34 tasks included in our elicitation study. Gestures ranked by agreement rating, with Vatavu and Wobbrock's high, medium, and low agreement categories indicated by color. Orange bars show agreement scores when gestures are placed within more general groups of type and length.

4.3.4 Vocabulary

We also examined the *gesture vocabulary* of participants, i.e. how many unique gestures were used within the set of gestures elicited from a participant. We define unique gestures as follows: If the gestures elicited by the participants were to be drawn on a piece of paper, two gestures would be considered unique if they are either: a) visually different, b) oriented differently, or c) constructed in a different order of strokes. As an example, a list of compound gestures: 'M', 'MV', 'VV', and 'MM' would only have two unique gesture symbols in the dictionary, 'M' and 'V'.

Alongside the gesture vocabulary size, we also examined the types of gesture used by participants. Based on an initial analysis of our data set, we characterized gestures as either ideographic (a shape or movement) or alphanumeric (a stylized letter or number). On average, the number of unique ideographic symbols per participant was 19.6 ($\sigma = 4.3$) across all 34 tasks. For all of the *filter* tasks, the participants naturally proposed either completely alphanumeric or a mixture of ideographic-alphanumeric gesture sequences. For example, the elicited gestures for the *Jump to Movie* task in the Netflix grid-based layout proposed either writing out the letters of the movie title, or by specifying the row and column number corresponding to its position in the grid. Overall, 65.6% of collected gestures were classified as ideographic, 25.9% as alphanumeric, and 8.53% comprised a combination of the two.

4.4 Personalization

When considering gesture sets, one question is how a set of gestures (e.g. 19 on average) can scale to support more than 30 different commands. Participants extended the vocabulary through gesture symbol concatenation. More specifically, when eliciting gestures from participants, participants would at times create a gesture command using one symbol from the vocabulary. However, it was also quite common for participants to create two and three gesture sequences for gesture input. We use the term 'conceptual length' to describe the number of atomic shapes used to create a gesture. We defined the concept of a shape as a series of curves and strokes being drawn in a specific order, at a specific location. If either the location or the order of the strokes is independent of the preceding stroke, then it marks as a natural delimiter among two atomic constructs. For example, if a participant designed a gesture that was 'a house shape' comprised of a triangle on top of a rectangle, the gesture would have a conceptual length of 1. This is because of the layout constraint of the triangle and rectangle. In contrast, a gesture that consisted of a triangle followed by a rectangle with no constraint on positioning would have conceptual length 2. As another example, a gesture sequence of 'ZI', contains two alphanumeric shapes that are only dependent on the order and not the location of the trace. Thus, this gesture has a conceptual length of 2. We binned every gesture as "Single", "Double" or "Three+" to denote the conceptual length of one, two, or more than three. Of the 34 tasks in our study, only in two of the tasks (*Zoom In/Zoom Out*) did participants consistently use a single symbols gesture. In all other cases, gestures elicited from participants were a mix of one-, two-, and three+ symbols.

4.5 Design Implications

While the elicitation study provides a significant set of data points, the goal is to leverage that data to provide guidelines for the design of at-the-side gesture sets. An overview of the design implications from our elicitation study are as follows:

- Support a sufficiently large gesture set. Users used, on average, approximately 20 unique gestures over the 34 tasks provided in this study. Given the restricted context of interacting with an external display, we believe that a sufficient gesture set can be comprised with 25 to 100 symbols.
- Consider user-specific versus designer-specified gesture sets. Overall, we see relatively low agreement scores across navigation, action, and filter task types. However,

at the same time, we see high recall rates for users per their own gestures, despite specifying gestures for over 30 tasks. We believe that the combination of low inter-participant agreement and high recall rates is a key characteristic of domains that can benefit from allowing users to create their own gesture libraries.

- Include both Ideographic and Alphanumeric gestures. We found a combination of both ideographic or abstract gesture sets and alphanumeric characters as symbols in participants' elicited gesture dialects. This is perhaps unsurprising: language is an important tool for expressing concepts, and, particularly in command-rich environments, text is often the best way to filter command lists. Consider, for example, invoking programs on modern operating systems, where short typed text strings filter command lists to a manageable length before users select from a list.

- Support multi-glyph gestures. One way that users naturally expand a command language is by combining symbols together into bigrams and trigrams; in our study, one-third of the hundreds of input gestures collected from participants were multi-symbol gestures. While support for bigrams and trigrams effectively expands the input language, it introduces challenges for symbol segmentation.

- Finally, we see a need for organic and dynamic input sequences, gesture traces semantically constructured for one-time use. In public contexts, one may only infrequently, perhaps only once, invoke a specific command. As an example, locating a specific datum on a high density information display might require filtering and selecting, but that datum might only be relevant in the present interaction. Symbols that can be effectively combined into novel, single-use commands have high value in public contexts.

Chapter 5

Evaluating Expanded Gesture Sets

5.1 Overview

In this chapter, we return to our prototype system and explore its ability to handle enlarged gesture vocabularies. We place an added focus on alphabetic characters via the Palm Graffiti gesture set as a means to provide an extensible set of long-form gestures for boosting the vocabulary size to be arbitrarily long.

One open question we, as researchers, struggle with when presented with results of elicitation studies that amount to design implications is an inability to discern whether or not the observations are actionable. How well will the guidelines above work using real-world gesture recognizers? Is user input sufficiently stable that user-elicited gesture sets are practical (i.e., can users train a classifier quickly on a gesture set)? How can we create multi-glyph gestures that can be reliably recognized? How much accuracy can we generate from n-gram-based input gestures?

5.2 Online Realtime Study

To evaluate side gesture interaction in more ecologically valid settings and to evaluate the potential of multi-glyph gesture recognition, we conducted a second laboratory study. As a first set of multi-glyph gestures, we used the most common bigrams and trigrams in English. Our system is designed specifically for contexts where gesture input is segmented

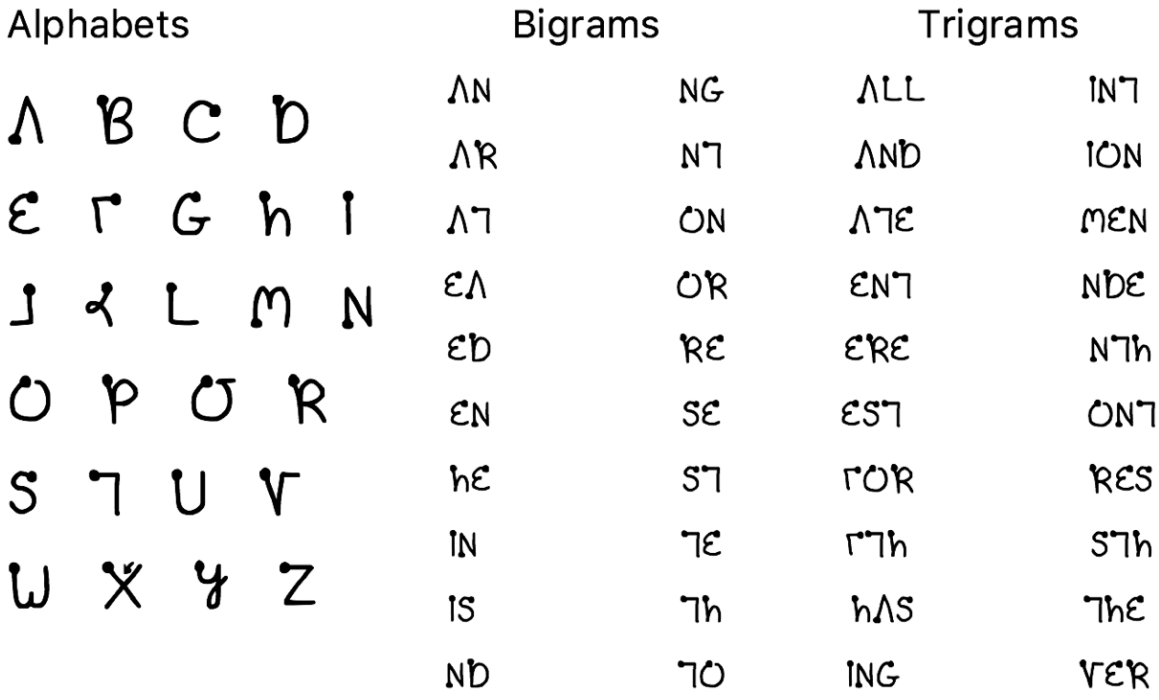

Figure 5.1: The Palm Graffiti (left) and $1 unistroke (right) gesture set containing 26 alphabet gestures. The bigram and trigram gesture sets, generated from the unistroke models, contained 20 gestures each.

from the everyday movement of a device via a clutch [75] or delimiter[63]. In this follow-up study, our system does not assume cleanly segmented data. Once data collection is started, we initially performed analysis on all input, including gesture and positional noise.

Sensor data was recorded for each individual trial as a continuous stream including noise and non-relevant motion. We perform segmentation of gestures (regardless of length) by detecting the idleness of the arm in the projected drawing plane. We use simple threshold-based segmentation to denote the beginning and end of a particular gesture. There are two components to the threshold, movement and time. We tuned our movement threshold from user input data from informal pilot studies. We use a threshold of 4 cm of arm movement in our horizontal plane, a displacement that seems to work well. From our initial pilot study, we found a temporal threshold of 1500 ms to be a good threshold

to distinguish short pauses between individual characters in a bigram or a trigram gesture and longer pauses between gestures. In summary, less than 4cm arm movement within 1.5 seconds was detected as a gesture segment. One single gesture could be a unigram character, a bigram character, or a trigram character.

Once the data was segmented and annotated as a gesture state, the recognizer performed DTW on the segmented data. Each sample was then annotated by the recognizer with the recognized gesture, trial instance information, and whether the prediction was successful or unsuccessful.

We recruited 12 participants (5 female) from a local university for a 30-minute session. All participants were right-handed, and were instructed to wear the smartwatch on their non-dominant arm during the study.

The procedure was as follows:

1. Participants were welcomed, provided background information on the purpose of the study, and signed an informed consent form. Next, participants were instructed on how to perform gestures using the smartwatch, asked to put the smartwatch on and adjust it for comfort.

2. Participants then trained the recognizer. During the training phase, each participant was asked to repeat each of the single character Graffiti gestures (A to Z) 4 times, for a total of $26 \times 4 = 104$ trials. To record clean sensor information for each motion gesture, participants delimited the start and end of each gesture, and were instructed to keep their arm idle when not performing a gesture. The collected data were used generate a template for the recognizer's dictionary as described in the previous section. All samples provided as training data were, however, logged and preserved so that we could perform post-hoc analysis of variations in recognizer behavior.

3. Next, participants completed the recognition phase of the experiment. The recognition phase simulated a single gesture session where the user would perform gestures segmented by idle arm movement. The recognition phase comprised three stages of increasingly complex gestures: first unigram, then bigram, and trigram. We structured the study this way so that participants could familiarize themselves with the initial unigram gesture set before attempting bigram and trigram gestures. During each stage, the experimental software prompted participants to perform a random gesture of the specified length. If the gesture was classified correctly, the system would provide on-screen feedback by highlighting the gesture in green. If the gesture was not recognized, the user was notified by red-colored feedback and was given one additional attempt to perform the same gesture. If the gesture was not recognized in either attempt, the trial was marked as an error and the participant proceeded to the next trial.

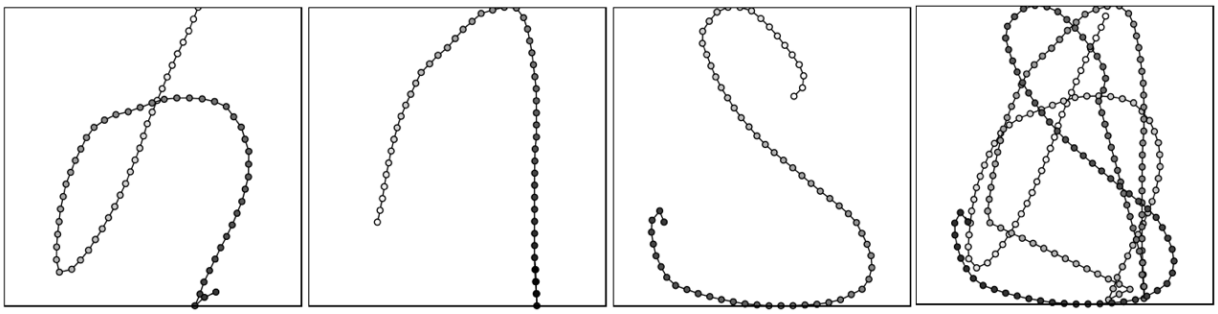

Figure 5.2: A template for a bigram or a trigram sequence is generated dynamically based on its individual unigram templates. The first three squares represent template for unigrams ‘H’, ‘A’, ‘S’. The last square represents the generated template for the trigram ‘HAS’.

5.2.1 Generating the Template-based Model

During our experiment, we included a training phase to collect samples of the 26 unigram gestures to seed the recognizer. The recognizer was trained by each participant so that the gestures being recognized were participant specific. No gesture traces were shown to the participants during both the training and the recognition phase as to not bias their natural behavior for performing gestures.

To create a set of gesture templates, we leveraged the DTW algorithm to find the best candidate input sample from every input gesture. For each training sample for a given gesture, we calculated its warp distance to all other training samples and then selected one single template to represent the gesture, the most representative of the candidate gestures, given by computing the minimum accumulative warp distance between every training sample collected for a single unigram.

To generate templates for the bigram and the trigram dictionary within our recognizer, we concatenated multiple unigram templates in order of the letter sequence. For example, to generate a template for the trigram ‘has’, we layered independent unigrams ‘h’, ‘a’, and ‘s’ (Figure 5.2, right). We also interpolated data points between the end point of one unigram and the start point of the next, thus creating a single unistroke template that represented a multicharacter sequence, e.g. a bigram or trigram. The benefit of this approach is that once the recognizer is trained on the 26-gesture unigram alphabet, concatenation can yield an input language of up to 726 bigrams and of over 17,000 trigrams without additional training.

5.2.2 Results

Using the unoptimized dynamic time warping recognition algorithm, we found a successful gesture input rate of 63% for unigrams, 65% for bigrams, and 70% for trigrams as the first attempt. The decrease in recognition accuracy from the pilot study can be attributed to the difference in segmentation of data in online recognition systems to user-segmented offline capture data.

Using fully logged input stream data, we performed a post hoc analysis of errors to identify how the un-optimized classifier's recognition rate might be improved. We found that two of our twelve participants were outliers, with recognition rates below 50%, and that these two users were significantly negatively impacting recognition rate. In some instances, it may be appropriate to identify and eliminate outliers from analysis; however we were interested in understanding why these participants were outliers and whether our algorithm could be refined to better recognize their gestures.

An initial analysis of outlier data highlighted an alignment issue associated with data points in template and candidate, a problem particularly acute for participants who gestured quickly and less carefully. When captured motion data was sparse, template and candidate data points would become shifted and the warp distance would be unusually large between a candidate gesture and all templates. To address this issue, we refined our classifier using the interpolation approach from the 1$ recognizer [78] to ensure an identical number of data points in the gesture template and candidate gesture. Using our archived training samples, we generated new normalized templates and then re-analyzed character input as a raw input stream to ensure as accurate a re-play of recognizer behavior as possible.

Overall, this change boosted recognition accuracy by approximately 20% for first instance success. We suspect that success rates on second and third attempts may be higher as well, but because this analysis was done post-hoc, it is impossible to determine how second and third attempts would be affected.

However, to ensure that we were not over-optimizing on our test data and to assess the potential of this improvement, we asked the poorest performing participant to repeat our study with the optimized recognizer. This participant initially had recognition rates of between 40% and 55% for each dictionary. In our follow-up test, the recognition rate for this participant was 86% for unigrams, 91% for bigrams, and 85% for trigrams.

Alongside variations in motor performance, our initial analysis of recognizer behavior specifically considers a single best recognition result. We also assume that, in every candidate instance, participants attempted to make a specific gesture, i.e. they did not

make an error (e.g., when prompted for 'I', they did not, accidentally, make an 'L'). However, during our observations of participants, we noted that due to the lack of familiarity with graffiti, participants would, in approximately 10% of input cases, begin an incorrect gesture. Once a participant realized their mistake, and knowing that they had two attempts, they would shake their hand or perform some other nonsense gesture, allow the classifier to misrecognize, and then try again. Obviously, this had an impact on the first-attempt accuracy in recognition, but not on overall accuracy.

The common way to deal with input slips where a user begins an incorrect gesture is to use thresholds that trade-off the cost of false positives versus false negatives to create a "reject" category [15]. We did perform a thresholding analysis using a variable reject threshold which allowed us to generate error rates of less than 4% with reject rates below 15%.

Finally, to optimize the recognition rate for gestures, one might improve accuracy by selecting gestures for the dictionary that are maximally distinct [57], thus decreasing the likelihood of collision. While these and other optimizations are possible, the goal of this study was not to optimize our recognizer for real-world use, but instead to study the feasibility of side-gesture input.

5.2.3 N-best Lists

We performed a post-hoc analysis of collected gesture data to understand whether the recognition rates could be achieved without developing a special-purpose recognizer. We processed the collected data streams by resampling data-points to correct for issues such as jitter and repositioning movements. With minor data processing and the DTW-based recognizer, mean recognition rates for the unigrams, bigrams, and trigrams were 82%($\sigma = 5.71\%$), 80% ($\sigma = 3.63\%$), and 82%($\sigma = 13.8\%$), respectively. We were also interested in the deviation of the incorrectly recognized gestures from the template model in terms of the predicted gesture.

Taking the best two candidates from the recognizer classification, with the first candidate being the predicted, we saw an increase in accuracy of the unigrams to 92% ($\sigma = 3.92\%$), bigrams to 92% ($\sigma = 5.95\%$), and the trigrams to 90% ($\sigma = 10.15\%$). The results are summarized in Figure 5.3. This means that even when the gesture was wrongly classified, the right prediction was usually the second guess. In designing such a system for real-world use, we therefore suggest that strategies such as n-best lists improve recognition rates by allowing user input to disambiguate between the top classification candidates through a prompt. Additional techniques such as rejection sampling, confusion matrices,

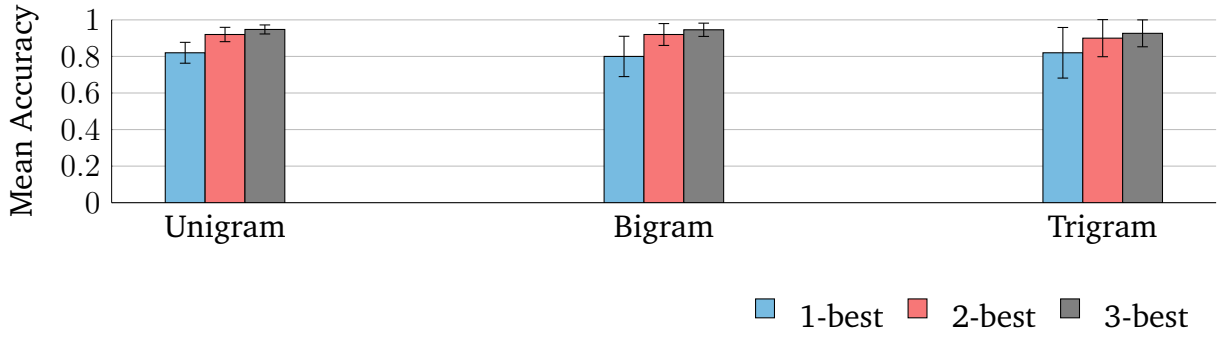

Figure 5.3: 1-best, 2-best, and 3-best recognition accuracy for unigrams, bigrams, and trigrams with optimized classifier.

bi-level thresholding, and adaptive recognizers could be applied to further improve such systems in practice. Figure 5.3 illustrates the accuracy rates using the n-best lists. This provides in important insight into the kinematic characteristics of the templates; due to the inherent nature of template-matching classifiers, the correct classification is highly likely to be from the top few closest candidates. This result encourages further exploration into techniques such as bi-level thresholding and feature extraction to further attenuate confusion.

5.2.4 Confusion Matrices

For analyzing conflicts and cases of high error, we generate confusion matrices for the alphanumeric $1 gesture set and the resulting bigram and trigram templates used for the study. Naturally, we expect that characters sharing similar visual properties will be primary candidates to cause confusion. In the following subsections, we explore the confusion matrices for each vocabulary set and pinpoint specific gestures causing incorrect classifications. Confusion matrices for unigram, bigram, and trigrams are shown in Table 5.2.4, Table 5.2.4, and Table 5.2.4 respectively.

For unigrams, we see minor classifier confusion for ‘A’ (‘M’), ‘D’ (‘P’), ‘I’ (‘T’), ‘O’ (‘U’), and ‘Q’ (‘G’), occasionally in both directions. The similarity in gesture movement and issues of full closure sensing due to imperfect data segmentation were the primary contributors to these errors. The distinctiveness of temporal data for these single-letter gestures is also relatively limited as the sheer number of datapoints on which the classifier operates causes a higher weight to be placed on the tail motion before and after the actual

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	23	0	0	0	0	0	0	2	2	0	0	0	5	0	0	0	0	1	0	2	0	0	0	0	1	0
B	0	29	0	0	0	0	0	1	1	0	0	0	1	0	0	3	0	1	0	0	0	0	0	0	0	0
C	0	0	31	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0
D	0	5	0	25	0	0	0	2	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0
E	0	0	3	0	32	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	1	0	0	28	0	0	6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	30	0	1	0	0	0	0	1	1	0	2	0	0	0	0	1	0	0	0	0
H	0	0	0	0	0	0	0	33	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	2	0	0	28	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
J	0	0	0	0	0	1	0	0	2	31	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
K	0	0	0	0	0	2	0	0	0	0	33	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	6	0	2	0	1	0	0	0	0	26	0	0	0	0	1	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	3	1	0	2	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	1	0	2	0	0	0
O	0	0	0	0	0	0	1	0	0	0	0	0	0	0	25	0	2	0	0	0	8	0	0	0	0	0
P	0	1	0	8	0	0	0	1	0	0	0	0	0	0	0	25	0	1	0	0	0	0	0	0	0	0
Q	0	0	1	0	0	0	6	0	0	0	0	0	0	0	3	0	23	0	0	0	2	0	1	0	0	0
R	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	0
S	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	34	0	0	0	0	0	0	0
T	0	0	0	0	0	1	0	0	4	1	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0
U	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	0	3	0	0	0	25	1	1	0	0	0
V	0	0	1	0	0	0	0	2	0	0	0	1	0	1	0	0	0	0	0	0	2	29	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	32	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0
Y	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34	0
Z	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32

Table 5.1: Confusion matrix for unigrams

gesture.

	AN	AR	AT	EA	ED	EN	HE	IN	IS	ND	NG	NT	ON	OR	RE	SE	ST	TE	TH	TO
AN	24	0	0	0	0	0	2	7	1	0	0	1	0	0	0	0	0	0	1	0
AR	0	23	1	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	9	0
AT	0	0	21	0	1	0	0	0	1	0	0	10	0	0	0	0	2	0	1	0
EA	0	0	0	31	2	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0
ED	0	0	0	0	32	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0
EN	0	0	0	0	0	23	0	13	0	0	0	0	0	0	0	0	0	0	0	0
HE	0	0	0	0	0	0	32	0	0	0	0	1	0	0	2	0	1	0	0	0
IN	1	0	0	0	0	1	0	31	0	0	0	0	0	0	3	0	0	0	0	0
IS	0	0	0	0	0	0	1	4	26	1	0	0	0	0	1	0	2	0	1	0
ND	0	0	0	0	2	0	0	0	1	30	0	0	0	0	1	0	0	0	2	0
NG	0	0	0	0	0	1	0	1	0	0	31	0	0	0	0	0	0	0	0	3
NT	0	0	0	0	0	0	0	0	1	0	0	33	0	0	1	0	0	0	1	0
ON	0	0	0	0	0	2	0	0	0	0	0	0	34	0	0	0	0	0	0	0
OR	0	0	0	0	0	0	0	0	0	0	1	0	4	30	0	0	0	1	0	0
RE	0	0	0	0	0	0	9	1	0	0	1	0	0	0	25	0	0	0	0	0
SE	0	0	0	0	0	0	4	0	1	0	0	0	0	0	0	25	1	5	0	0
ST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	33	0	0	0
TE	0	0	0	0	0	0	2	0	0	0	2	1	0	0	0	0	1	29	1	0
TH	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0
TO	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2	1	30

Table 5.2: Confusion matrix for bigrams

For bigrams and trigrams, the most commonly confused symbol was ‘EN’ (‘IN’)/‘ENT’ (‘INT’). This confusion is slightly harder to diagnose; we hypothesize that, because the

	ALL	AND	ATE	ENT	ERE	EST	FOR	FTH	HAS	ING	INT	ION	MEN	NDE	NTH	ONT	RES	STH	THE	VER
ALL	20	0	3	0	0	0	0	1	0	0	3	0	0	1	0	0	0	1	7	0
AND	0	27	0	0	0	0	0	1	1	1	0	0	0	0	3	0	0	3	0	0
ATE	0	0	30	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5	0
ENT	0	0	0	19	0	0	0	1	0	0	14	0	0	0	1	0	0	0	0	1
ERE	0	0	0	0	28	0	0	0	0	1	3	0	0	0	0	0	0	0	4	0
EST	0	0	0	1	0	29	0	2	0	0	1	2	0	0	0	0	1	0	0	0
FOR	0	0	0	0	0	0	30	0	0	0	0	3	0	0	0	0	0	0	0	3
FTH	0	0	0	1	0	0	0	28	0	0	1	0	0	0	5	0	0	1	0	0
HAS	0	0	0	0	0	0	0	2	29	0	4	0	0	0	1	0	0	0	0	0
ING	0	0	0	0	1	0	0	1	0	30	2	1	0	0	0	0	0	0	1	0
INT	0	0	0	0	1	0	0	1	0	0	33	0	0	0	0	0	0	0	0	1
ION	0	0	0	0	0	0	0	3	0	0	0	33	0	0	0	0	0	0	0	0
MEN	0	0	0	0	0	0	0	1	0	0	0	1	34	0	0	0	0	0	0	0
NDE	0	0	1	0	0	0	0	0	0	0	1	0	0	28	0	0	0	0	6	0
NTH	0	0	0	0	0	0	0	3	0	0	2	0	1	1	29	0	0	0	0	0
ONT	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0
RES	0	1	0	0	0	0	0	2	1	0	0	0	0	0	1	0	30	0	0	1
STH	0	0	0	0	0	0	0	3	0	0	1	0	1	0	0	0	0	31	0	0
THE	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	35	0
VER	0	0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0	0	0	32

Table 5.3: Confusion matrix for trigrams

suffix of the gesture is identical, the temporal warping of points may push segmentation to a higher match in some cases, particularly when the 'E' gesture is made quickly.

Overall, some of the classification inaccuracies visible in the confusion matrix can be attributed to user variation in performing the gesture and the tendency to adjust or 'clutch' before initiating the gesture. Nonetheless, analyzing pair-wise similarity between each gesture in the vocabulary helps improve recognition; if specific cases for confusion are known well ahead of the recognition task, the system can choose to present the user with a menu selection interface to disambiguate the gesture.

5.2.5 Increasing Vocabulary Size

Up to this point in our study, our focus has been on gesture vocabularies that are restricted in size. However, as we note, the use of bigram and trigram gesture sets permits significantly larger dictionaries of gestures. Bigrams support 676 unique gestures; trigrams in excess of 17,000 unique gestures.

Because our classifier recognizes multistroke gestures by automatically generating bigram and trigram templates through the concatenation of unigrams, we can easily observe the effect of larger alphabets. Figure 5.4 shows how accuracy rate declines as dictionary size increases. Bigram gestural languages, in particular, seem very resilient to increases in the size of the gesture dictionary, especially when n-best lists are leveraged. With a dictionary size of 80, bigram recognition still results in 3-best recognition accuracy of over 80%.

It should also be noted that we performed no optimization on our dictionary of gestures

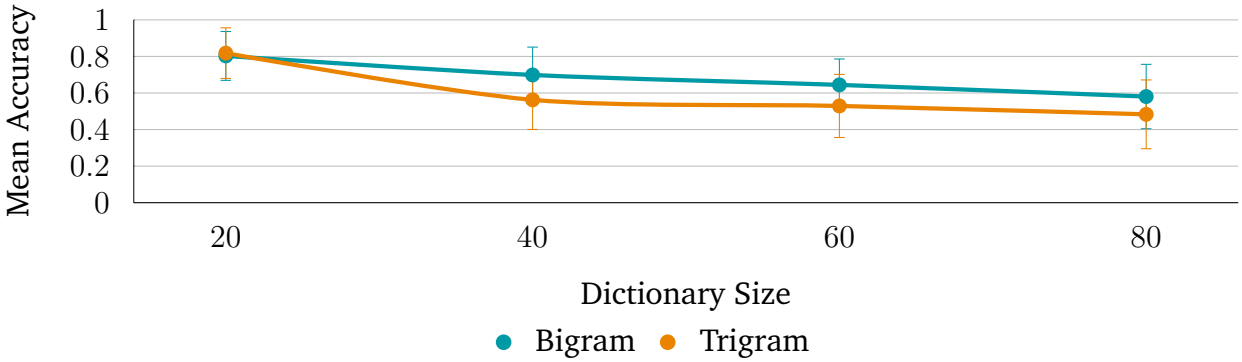

Figure 5.4: Recognition accuracy decline as dictionary size increases, for both bigrams and trigrams

to perform this analysis. In an ideal world, when designing a gesture language, one might improve accuracy by selecting gestures for the dictionary that are maximally distinct, thus decreasing the likelihood of collision.

5.2.6 Candidate Similarity

Adjustments in thresholds can optimize the error-reject tradeoff. The classifier picks a candidate that has the minimum computed warping distance as computed by DTW. However, confusion increases as the difference in computed warping distance for multiple candidates becomes small. This informs our metric of similarity index: the ratio of raw DTW warping distance for two gesture candidates. The reasoning for increasing rejection rates with increased confusion is grounded with confidence intervals for pattern matching algorithms; we estimate the level of confidence for the classifier based on the similarity ratio of the top candidates. Having a high similarity ratio means the recognizer cannot disambiguate between the top candidates and is unable to classify with high confidence.

To determine the effect of ambiguity on recognition, we explored a simple confidence test. As a confidence value, we use the ratio of the warp distance between C and the closest template, T_1, and the warp distance between C and its second choice template, T_2, i.e., Confidence = T_1/T_2. Because T_1 always has the minimum warp distance, this confidence interval will always be less than 1.0. However, the closer the two warp distances are to each other, the closer this value will be to 1.0. We call this ratio the *Similarity Measure* for a candidate gesture C, i.e. a measure of the similarity in proximity of its closest- and second-closest template.

	Threshold	1.0	0.95	0.9	0.85	0.8	0.75	0.7
Unigram	Accuracy	0.82	0.85	0.87	0.88	0.89	0.91	0.92
	Reject rate	0.00	0.04	0.09	0.13	0.17	0.22	0.25
Bigram	Accuracy	0.80	0.83	0.87	0.90	0.93	0.95	0.96
	Reject rate	0.00	0.08	0.15	0.22	0.30	0.38	0.44
Trigram	Accuracy	0.82	0.85	0.89	0.92	0.95	0.96	0.97
	Reject rate	0.00	0.08	0.15	0.23	0.31	0.39	0.48

Table 5.4: The effect of thresholds on rejection rates and mean accuracy.

For the unigram, bigram, and trigram sets, Figure 5.5 plots similarity measures with correct recognition results shaded in light blue and incorrect recognition results shaded in red. The contrast in similarity scores is quite striking. Recognition errors are clustered to the right, where the ratio of closest to second closes template approaches 1.0.The similarity measure for each template can be used to threshold recognition and define a third option for the classifier, a reject rate [15]. Table 5.4 shows the effect of thresholds on reject rates and accuracy. As shown, as thresholds increase, we can boost recognition rates significantly over 90%, but at the cost of significant rates of rejection or false negatives. An optimal cut-off for thresholding depends on the relative cost of error versus false negative and will vary from domain to domain.

5.2.7 Fully Generated Template Models

In our evaluation of gestural input accuracy, we included a training phase where participants performed unigram gestures, and the system used the elicited unigram gestures to generate a template library for the recognizer against which unigrams, bigrams, and trigrams were matched. However, recognizer training can be tedious for users. One might then ask whether recognizers can be trained with generic templates.

To estimate recognition accuracy without training for user adaptation, we have tested each participant's data against the model made out of the other participant's data. Figure 5.6 shows recognition accuracy using leave-one-out cross-validation. Overall, there was no significant difference between recognition accuracy using user-dependent model and accuracy using general model.

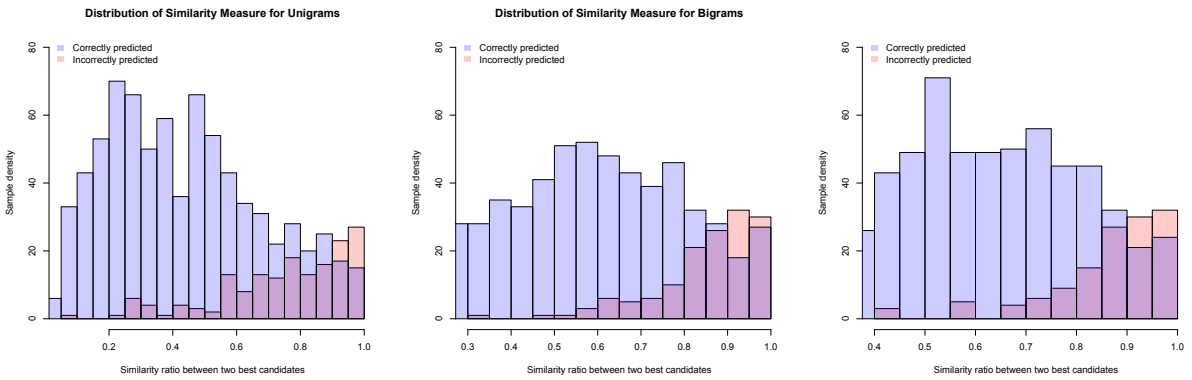

Figure 5.5: Candidate similarity ratios for unigram, bigram, and trigram template classifications

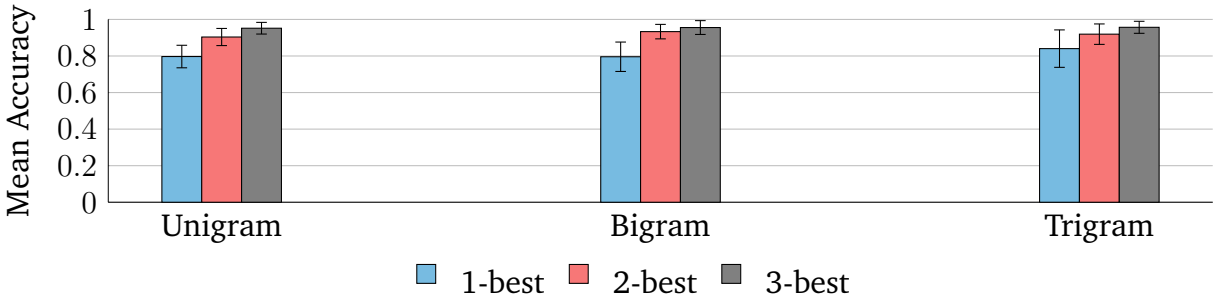

Figure 5.6: 1-best, 2-best, and 3-best recognition accuracy for unigram, bigram, and trigram template models generated through leave-one-out cross-validation where a participant's test data was tested against model data generated from 44 templates excluding that participant's data.

We also performed preliminary exploration into programmatically generated template models, avoiding user training altogether. Given that our gestures are performed in an $[x, y]$ plane, and given that our unistroke unigrams have a pre-specified drawing order, we can sample a two-dimensional static drawing of an input candidate, generate a template from that data, and use that template within our library. Figure 5.8 shows the training dataset generated from the fonts and the users. Figure 5.7 shows this accuracy for unigrams, bigrams, and trigrams using 1-best, 2-best, and 3-best lists with off-line, automatic template generation. Of note is that 3-best lists for generic templates generate recognition accuracies of 84% for unigrams, 89% for bigrams, and 88% for trigrams

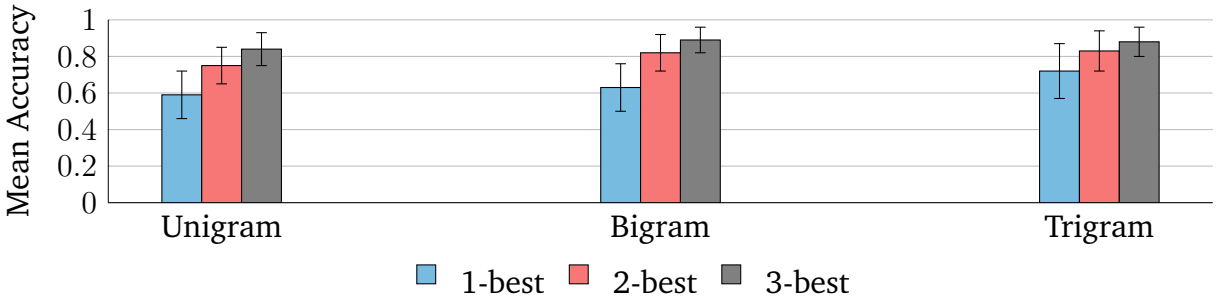

Figure 5.7: 1-best, 2-best, and 3-best recognition rates for an automatically generated unigram, bigram, and trigram template model.

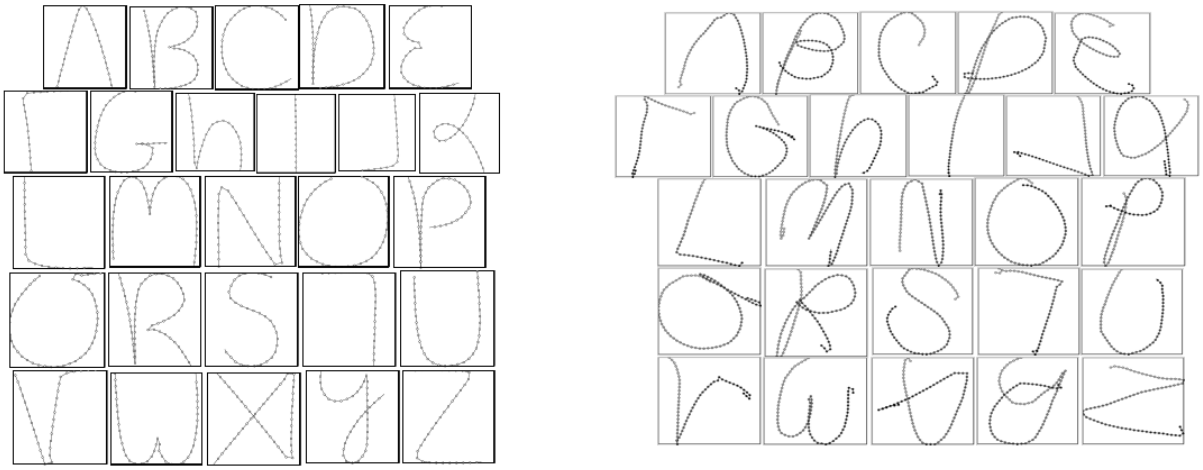

Figure 5.8: Training dataset generated from vectorized unistroke fonts (left) and from the user training samples (right).

without the need for user training. This accuracy could be boosted even higher using reject rates: note that we are not performing any confidence thresholding in this analysis, and, in our initial data set, approximately 10% of user actions were a result of slips.

Overall, there is a dropoff in accuracy as the generic templates do not perform as well as user-specific templates elicited during training. However, in real-world deployments, one could imagine these generic templates as a bootstrapping mechanism for gestural input. New users to the system could perform reasonably – for example with 90% accuracy using 3-best lists – until a user-tuned set of templates can be elicited from their input data

to support higher recognition.

5.3 Discussion

One limitation of this system is that our recognition algorithm, dynamic time warping, has well-known limitations of nearest neighbor classification on noisy input data compared to techniques such as hidden markov models; nearest neighbor template classification has more frequently been proposed as a recognition tool [78]. However, dynamic time warping is a tractable, easily implemented algorithm, and so for prototyping and rapid deployments, these results are invaluable to researchers and designers. For more sophisticated deployments, more sophisticated recognition strategies and priors from context hold the potential to attain the more accurate recognition rates required in tasks where both errors and false negatives are inadvisable. From the perspective of bigram and trigram generation, our creation of longer gestural sequences can be formalized further. More sophisticated algorithms would incorporate real-world training on these more extensive languages. However, as our interest is overall feasibility, we thought that an initial foray that examined how gestures can be combined, segmented, and recognized even with ambiguous segmentation was a worthwhile exploration given our desire for larger gestural languages. Together, these limitations should highlight the fact that the evaluation results are pessimistic estimates of the overall accuracies possible for a real-world system.

The implications for a system's ability to recognize a linguistic vocabulary are huge; users can form better associations with textual gestures similar to the way text-based user interfaces (TUIs) such as the Windows command prompt provide a powerful list of operations to the user. The area of making gesture-based interfaces more natural and forgiving has been hugely underexplored. Current gesture-based systems are analogous to automated telephone machines waiting for the user to provide highly specific input. Allowing a gesture system to encapsulate a much larger vocabulary, and thus overload one action with multiple gesture words, can go a long way in building *gestural* interfaces and a potential substitute for modern voice-based assistants such as Siri.

We believe that the onus of usable gestural systems lies on both the designer and the user: the designer must allow a framework of expressivity and creativity which the user can take advantage of and personalize to suit their needs. The *WatchTrace* system serves as an example of a prototype built under the at-your-side paradigm to allow for gesture 'words' to be formed by relatively subtle and quick arm motions. As the designer of the gesture set provides a set of atomic template models, the system can then generate new

vocabulary by learning the requirements of the end-user and improving its classification over time. We also consider, within reach, the possibility of the users designing their very own gesture templates and making systems even more expressive.

Chapter 6

Conclusion and Future Work

In this chapter, we present a summary of the design and feasibility exploration of at-your-side gestural interaction described in this thesis: the exploration of the design space of arm-based motion gestures, the design motivations for developing an at-your-side gesture paradigm, and feasibility of such a system. We then suggest areas where this work can be extended in the future either by optimizing the current recognition system or by adopting alternative techniques to improve the design and usability of this gesture space.

6.1 Conclusion

This thesis first described the design motivation of a fatigue-minimizing gesture paradigm using an arms-down posture while providing a rich and extensive gesture vocabulary. We surveyed the domain of arm gesture interaction to identify unexplored but prevalent challenges for improving the usability of such systems. Informed by physical effort metrics such as *Consumed Endurance* while performing mid-air gestures, moving the plane of interaction from in-front to at-side of the user's body can not only reduce the exertion but also provides a form of socially acceptable gesture input.

Informed by the elicitation study to understand the users' the mental model and level of comfort in at-your-side gesture paradigm, there is a largely unsolved need for gesture systems to be rich, supporting a high volume of abstract and alphanumeric gestures; and personalizable, allowing for gesture concatenation and chaining of input for complex workflows. Even though this work infers a low-to-moderate consensus for a standard

gesture set, the study demonstrates promise with high user satisfaction and mnemonic association.

The evaluation of a prototype system in this paradigm, using commodity hardware and recognition schemes, show promise into the feasibility of expressive, conversational gesture interfaces. With four training samples, an effective recognition of over 90% is feasible, by incorporating standard error-thresholding techniques such as n-best lists, in a real-time recognition system that users train themselves, and generic template sets (collected from other users) also achieve high recognition accuracy. Alongside this, we demonstrate a technique for concatenation of simple gestures into bigram and trigram gestures and show high recognition accuracy for these more complex gestures. By providing a limited set of gesture models, these systems can be programmatically extended to form gesture chains, such as bigrams and trigrams, and adopt different dictionaries, abstract traces, geometric shapes, or alphanumeric characters, to match the context or environment.

At-your-side gestural input presents significant advantages for many persisting challenges associated with such types of freehand gestures. It is low-effort, as the gestures are performed in an arms-down posture. Delimiters can be leveraged to discriminate input from everyday movement. It is easily sensed using commodity wearable devices, simplifying deployment. The gesture dialect can is further extended thanks to the ability to leverage bi-grams and tri-grams. It can easily be combined with pointing through mode switches or by raising an arm to point at devices then lowering the arm to issue commands. Users rate the input as consistently low-effort and low-embarrassment for public use. Overall, we hope this exploration and evaluation of this paradigm can guide a more usable and considerate design of gesture-based systems.

In summary, this thesis has demonstrated that at-your-side gesture interaction is a viable, low-effort, expressive interaction paradigm for gesture-based input. The target of input systems can include wearables such as smartwatches or head-mounted displays, personal displays on smartphones and tablets owned by the user, and external displays found in the environment. Beyond display-based interaction, due to the expressiveness of large vocabulary input, side gestures merit consideration for control of an internet-of-things world.

6.2 Future Work

Through results and feedback from the study, we identify potential areas to explore and improve both the at-your-side paradigm and the gesture interface design. We present potential areas of further research worth investigating in the form of open questions:

1. Can we support the thousand most common verbs with reliable accuracy? As the feasibility section shows promise for extended dictionary lengths, a probe aimed at exploring the most common set of verbs in one or more languages could open up the potential for gesture systems to replicate the functionality of voice-based interfaces and reduce the system's learning curve.

2. Can we resolve gesture errors through candidate analysis and other factors or contexts? Highly similar gestures such as "I" and "J" are challenging for systems to distinguish. However, a directed study could be performed to build dictionary sets from different gesture combinations and estimate the level of confusion in a given gesture set. The study can be extended by introducing additional dimensions such as recently used gestures and frequently used gesture orders for practical improvements to accuracy.

3. Can the recognizer move completely on the watch? A complete disconnection of the watch with data processing systems, by performing both sensing and classification on-site, can open up possibilities for complex interaction where other currently internet-reliant techniques, such as voice recognition, are unavailable.

4. Can audio and vibrotactile feedback improve eyes-free communication? As a way to communicate the system state to the user and aiding the recognizer, specific on-site and off-site feedback mechanisms could be explored to shape and learn user behavior.

References

[1] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. Using mobile phones to write in air. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 15–28, New York, NY, USA, 2011. ACM.

[2] Jason Alexander, Teng Han, William Judd, Pourang Irani, and Sriram Subramanian. Putting your best foot forward: Investigating real-world mappings for foot-based gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1229–1238, New York, NY, USA, 2012. ACM.

[3] Tansu Alpcan, Sinan Kesici, Daniel Bicher, M. Kivanç Mihçak, Christian Bauckhage, and S. Ahmet Çamtepe. A lightweight biometric signature scheme for user authentication over networks. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, SecureComm '08, pages 33:1–33:6, New York, NY, USA, 2008. ACM.

[4] Alphabet Inc. Nest, 2017.

[5] C. Anslow, P. Campos, and J. Jorge. *Collaboration Meets Interactive Spaces*. Springer International Publishing, 2017.

[6] Apple Computer Inc. Apple human interface guidelines, 2016.

[7] Alec Azad, Jaime Ruiz, Daniel Vogel, Mark Hancock, and Edward Lank. Territoriality and behaviour on and around large vertical publicly-shared displays. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 468–477, New York, NY, USA, 2012. ACM.

[8] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[9] R A Bolt. “Put-that-there”: Voice and gesture at the graphics interface, 1980.

[10] G Borg. Psychophysical scaling with applications in physical work and the perception of exertion. *Scandinavian journal of work, environment & health*, 16 Suppl 1:55–8, 1990.

[11] Sebastian Boring, Marko Jurmu, and Andreas Butz. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, pages 161–168, New York, NY, USA, 2009. ACM.

[12] Jessica R. Cauchard, Jane L. E, Kevin Y. Zhai, and James A. Landay. Drone & me: An exploration into natural human-drone interaction. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp ’15, pages 361–365, New York, NY, USA, 2015. ACM.

[13] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, pages 3403–3414, New York, NY, USA, 2016. ACM.

[14] X A Chen, T Grossman, and D J Wigdor. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proceedings of the . . .* , 2014.

[15] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1):41–46, 1970.

[16] A Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 82–89. IEEE Comput. Soc.

[17] Gabriele Costante, Lorenzo Porzi, Oswald Lanz, Paolo Valigi, and Elisa Ricci. Personalizing a smartwatch-based gesture interface with transfer learning. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*, pages 2530–2534. IEEE, 2014.

[18] Nasser H Dardas and Nicolas D Georganas. Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3592–3607.

[19] Artem Dementyev and Joseph A Paradiso. Wristflex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 161–166. ACM, 2014.

[20] Ali Farghaly and Khaled Shaalan. Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):14:1–14:22, December 2009.

[21] A Ferscha, P Lukowicz, and F Zambonelli. The superorganism of massive collective wearables. In *Proceedings of the 2014 ACM . . .* , 2014.

[22] Adam Fourney and Richard Mann. Non-accidental features for gesture spotting. In *Computer and Robot Vision, 2009. CRV'09. Canadian Conference on*, pages 116–123. IEEE, 2009.

[23] D Goldberg and C Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT'93 and CHI' . . .* , 1993.

[24] Saikat Gupta, Sujin Jang, and Karthik Ramani. Puppetx: A framework for gestural interactions with user constructed playthings. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 73–80, New York, NY, USA, 2014. ACM.

[25] Lars Kai Hansen, Christian Liisberg, and Peter Salamon. The error-reject tradeoff. 1995.

[26] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3653–3656, New York, NY, USA, 2015. ACM.

[27] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1063–1072, New York, NY, USA, 2014. ACM.

[28] S Houben and N Marquardt. Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications. In *Proceedings of the 33rd Annual ACM . . .* , 2015.

[29] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *CHI*, 2017.

[30] Ankit Kamal, Yang Li, and Edward Lank. Teaching motion gestures via recognizer feedback. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 73–82, New York, NY, USA, 2014. ACM.

[31] Keiko Katsuragawa, Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 128–135, New York, NY, USA, 2016. ACM.

[32] Frederic Kerber, Philipp Schardt, and Markus Löchtefeld. Wristrotate: A personalized motion gesture delimiter for wrist-worn devices. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, pages 218–222, New York, NY, USA, 2015. ACM.

[33] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM, 2012.

[34] Jonghwa Kim, Stephan Mastnik, and Elisabeth André. EMG-based Hand Gesture Recognition for Realtime Biosignal Interfacing. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 30–39, New York, NY, USA, 2008. ACM.

[35] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. The gesture watch: A wireless contact-free gesture based wrist interface. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 15–22. IEEE, 2007.

[36] Sven Kratz, Michael Rohs, and Georg Essl. Combining acceleration and gyroscope data for motion gesture recognition using classifiers with dimensionality constraints. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 173–178, New York, NY, USA, 2013. ACM.

[37] Sven G Kratz and Maribeth Back. Towards Accurate Automatic Segmentation of IMU-Tracked Motion Gestures. In *CHI Extended Abstracts*, 2015.

[38] Gierad Laput, Robert Xiao, Xiang'Anthony' Chen, Scott E Hudson, and Chris Harrison. Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 389–394. ACM, 2014.

[39] Gierad Laput, Robert Xiao, and Chris Harrison. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 321–333, New York, NY, USA, 2016. ACM.

[40] Leap Motion, Inc. Leap Motion, 2017.

[41] Sang-Su Lee, Jeonghun Chae, Hyunjeong Kim, Youn-kyung Lim, and Kun-pyo Lee. Towards More Natural Digital Content Manipulation via User Freehand Gestural Interaction in a Living Room. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 617–626, New York, NY, USA, 2013. ACM.

[42] Y Li. Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human . . .* , 2010.

[43] Hai-Ning Liang, Cary Williams, Myron Semegen, Wolfgang Stuerzlinger, and Pourang Irani. User-defined surface+motion gestures for 3d manipulation of objects at a distance through a mobile device. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction*, APCHI '12, pages 299–308, New York, NY, USA, 2012. ACM.

[44] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[45] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 63–71, New York, NY, USA, 2015. ACM.

[46] Allan Christian Long, Jr., James A. Landay, and Lawrence A. Rowe. Implications for a gesture design tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 40–47, New York, NY, USA, 1999. ACM.

[47] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. Empress: Practical hand gesture classification with wrist-mounted emg and pressure sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2332–2342. ACM, 2016.

[48] Microsoft Corporation. Kinect for Xbox One, 2017.

[49] Danial Moazen, Seyed A Sajjadi, and Ani Nahapetian. Airdraw: Leveraging smart watch motion sensors for mobile human computer interactions. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 442–446. IEEE, 2016.

[50] Calkin S Montero, Jason Alexander, Mark T Marshall, and Sriram Subramanian. Would You Do That?: Understanding Social Acceptance of Gestural Interfaces. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 275–278, New York, NY, USA, 2010. ACM.

[51] Meredith Ringel Morris. Web on the wall: Insights from a multimodal interaction elicitation study. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 95–104, New York, NY, USA, 2012. ACM.

[52] Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface 2010*, GI '10, pages 261–268, Toronto, Ont., Canada, Canada, 2010. Canadian Information Processing Society.

[53] Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1099–1108, New York, NY, USA, 2013. ACM.

[54] Matei Negulescu, Jaime Ruiz, and Edward Lank. A recognition safety net: Bi-level threshold recognition for mobile motion gestures. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 147–150, New York, NY, USA, 2012. ACM.

[55] Matei Negulescu, Jaime Ruiz, Yang Li, and Edward Lank. Tap, swipe, or move: Attentional demands for distracted smartphone input. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 173–180, New York, NY, USA, 2012. ACM.

[56] Donald A Norman. Cognitive engineering. *User centered system design: New perspectives on human-computer interaction*, 3161, 1986.

[57] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[58] Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. Smartcasting: A discount 3d interaction technique for public displays. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, OzCHI '14, pages 119–128, New York, NY, USA, 2014. ACM.

[59] Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Tiltcasting: 3d interaction on large displays using a mobile device. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 57–62, New York, NY, USA, 2015. ACM.

[60] Lorenzo Porzi, Stefano Messelodi, Carla Mara Modena, and Elisa Ricci. A Smart Watch-based Gesture Recognition System for Assisting People with Visual Impairments. In *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*, pages 19–24, New York, NY, USA, 2013. ACM.

[61] S Ransalu and S Kumarawadu. A robust vision-based hand gesture recognition system for appliance control in smart homes. In *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, pages 760–763, August 2012.

[62] F. Ritter, C. Hansen, V. Dicken, O. Konrad, B. Preim, and H. o. Peitgen. Real-time illustration of vascular structures. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):877–884, Sept 2006.

[63] Jaime Ruiz and Yang Li. Doubleflip: A motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2717–2720, New York, NY, USA, 2011. ACM.

[64] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 197–206, New York, NY, USA, 2011. ACM.

[65] Jaime Ruiz and Daniel Vogel. Soft-Constraints to Reduce Legacy and Performance Bias to Elicit Whole-body Gestures with Low Arm Fatigue. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3347–3350, New York, NY, USA, 2015. ACM.

[66] H Sakoe and S Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978.

[67] Matt Sundstrom. Rock, Paper, Scissors; Reviewing the Myo Gesture Control Armband, February 2015.

[68] M Tang. Recognizing hand gestures with microsoft's kinect. *Palo Alto: Department of Electrical Engineering of . . . ,* 2011.

[69] Thalmic Labs. Myo Gesture Control Armband.

[70] Pham Quoc Thang, Nguyen Duc Dung, and Nguyen Thanh Thuy. A Comparison of SimpSVM and RVM for Sign Language Recognition. In *the 2017 International Conference*, pages 98–104, New York, New York, USA, 2017. ACM Press.

[71] Radu-Daniel Vatavu and Jacob O. Wobbrock. Formalizing agreement analysis for elicitation studies: New measures, significance test, and toolkit. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1325–1334, New York, NY, USA, 2015. ACM.

[72] D Vogel and R Balakrishnan. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of the 17th annual ACM . . . ,* 2004.

[73] James R Wallace, Nancy Iskander, and Edward Lank. Creating Your Bubble: Personal Space On and Around Large Public Displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2087–2092, New York, NY, USA, 2016. ACM.

[74] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3847–3851, New York, NY, USA, 2016. ACM.

[75] Daniel Wigdor and Dennis Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011.

[76] Julie R Williamson, Stephen Brewster, and Rama Vennelakanti. Mo!Games: Evaluating Mobile Gestures in the Wild. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 173–180, New York, NY, USA, 2013. ACM.

[77] Anusha Withana, Roshan Peiris, Nipuna Samarasekara, and Suranga Nanayakkara. zsense: Enabling shallow depth gesture recognition for greater input expressivity

on smart wearables. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3661–3670, New York, NY, USA, 2015. ACM.

[78] J O Wobbrock, A D Wilson, and Y Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM . . .*, 2007.

[79] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. ACM.

[80] Ying Wu, Thomas S Huang, and N Mathews. Vision-Based Gesture Recognition: A Review. In *Lecture Notes in Computer Science*, pages 103–115. Springer, 1999.

[81] Anbin Xiong, Yang Chen, Xingang Zhao, Jianda Han, and Guangjun Liu. A novel HCI based on EMG and IMU. In *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2653–2657. IEEE, 2011.

[82] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, pages 9–14, New York, NY, USA, 2015. ACM.

[83] Yixin Zhao, Parth H Pathak, Chao Xu, and Prasant Mohapatra. Demo: Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 471–471, New York, NY, USA, 2015. ACM.