Design and Evaluation of a Presentation Maestro

Controlling Electronic Presentations Through Gesture

by

Adam Fourney

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, 2009

I hereby declare that I am the sole author of this 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.

Abstract

Gesture-based interaction has long been seen as a natural means of input for electronic presentation systems; however, gesture-based presentation systems have not been evaluated in real-world contexts, and the implications of this interaction modality are not known. This thesis describes the design and evaluation of Maestro, a gesture-based presentation system which was developed to explore these issues. This work is presented in two parts. The first part describes Maestro's design, which was informed by a small observational study of people giving talks; and Maestro's evaluation, which involved a two week field study where Maestro was used for lecturing to a class of approximately 100 students. The observational study revealed that presenters regularly gesture towards the content of their slides. As such, Maestro supports several gestures which operate directly on slide content (e.g., pointing to a bullet causes it to be highlighted). The field study confirmed that audience members value these content-centric gestures. Conversely, the use of gestures for navigating slides is perceived to be less efficient than the use of a remote. Additionally, gestural input was found to result in a number of unexpected side effects which may hamper the presenter's ability to fully engage the audience.

The second part of the thesis presents a gesture recognizer based on discrete hidden Markov models (DHMMs). Here, the contributions lie in presenting a feature set and a factorization of the standard DHMM observation distribution, which allows modeling of a wide range of gestures (e.g., both one-handed and bimanual gestures), but which uses few modeling parameters. To establish the overall robustness and accuracy of the recognition system, five new users and one expert were asked to perform ten instances of each gesture. The system accurately recognized 85% of gestures for new users, increasing to 96% for the expert user. In both cases, false positives accounted for fewer than 4% of all detections. These error rates compare favourably to those of similar systems.

Acknowledgements

This thesis is the product of collaboration and cooperation; there are so many individuals from whom I owe thanks.

First, I would like to thank my supervisor and mentor Dr. Richard Mann. Dr. Mann provided me with much guidance, but also much freedom to explore and discover research topics that captivated my interest.

I also owe special thanks to Dr. Michael Terry. Dr. Terry introduced me to the field of human-computer interaction, and provided an immense amount of guidance and support throughout all stages of this research.

I would also like to thank Dr. Pascal Poupart. Many core concepts, regarding machine learning, I learned from you.

Finally, I would like to thank all those who volunteered to participate in Maesto's trials. These individuals agreed to wear brightly colored gloves and gesture wildly while being filmed. I am in your debt.

Dedication

I dedicate this thesis to Laura, my loving wife. In every task worth doing, you have supported me. None of this would have been possible without you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Electronic presentation systems, such as Microsoft PowerPoint [37], are designed to enhance one's ability to effectively communicate to a group of individuals. These systems have become invaluable tools for delivering presentations in a professional setting. Ian Parker, of *The New Yorker*, has reported that an average of 30 million presentations are given each day [43]. While this statistic is difficult to independently verify, Microsoft has reported that PowerPoint has an installed user-base of 500 million individuals [36]. In either case, it is clear that electronic presentation software plays an important role in the lives of millions of individuals.

Numerous research efforts have explored ways of improving presentation technology. For example, Palette [40] provides a tangible interface to presentations, enabling presenters to use barcoded cue cards to randomly access slides in a presentation; Slithy [70] explores how animations can be better utilized during talks; and MultiPresenter [26] explores how presentation software can leverage multiple projectors that are often available in large lecture halls. In this document, we are most concerned with improving how presenters *interact* with presentation software.

Electronic presentation systems are considered by many to be ideal candidates for gesture-based interaction. This is, in part, because presenters habitually gesture to the presentation screen while speaking [5, 40, 26]. Indeed, gesture-based input promises to seamlessly integrate with existing presentation practices. Baudel *et al.* note that, when using gestures to control a presentation, "most gestures are actually performed at the limit of consciousness", so that commands can be issued without much effort [2]. Similarly, Sukthankar *et al.* suggest that this form of interaction feels natural because it treats the computer "as if it were a member of the audience" [55]. Furthermore, research by Cao *et al.* found that a gestural interface encourages presenters to adopt a "more personalized, humanized, story-telling style", leading to more interactive and engaging presentations [5].

1.1 Open Issues

The literature contains many examples of gesture-based presentation systems [59, 54, 2, 7, 28]. While these systems demonstrate the possibility of gesture-based input to a slideshow, they have not been evaluated in real-world contexts. In fact, previous research often fails to consider how audiences respond to gestural control of presentations. Instead, evaluations have typically focused on the accuracy of gesture recognition (e.g., [2, 28]) rather than on the overall usability and usefulness of the system. It is therefore unknown if the real-world deployment of gesture-based presentation systems will elicit the advantages predicted in the literature.

Additionally, previous gesture-based presentation systems have often grafted gestural control onto existing presentation software, such as PowerPoint [28] or HyperCard [2]. In these cases, gestures are typically relegated to common navigational commands such as moving between slides. These navigation gestures are independent of slide content, and are often performed between talking points rather than co-occurring with speech. This can be contrasted with the gestures that spontaneously occur during a presentation; spontaneous gestures serve communicative purposes, co-occur with speech, and tend to be highly contextualized by the content and layout of the slides. Thus, while these natural gestures are often offered as motivations for gesture-based systems, they are qualitatively different than the navigation gestures which are actually implemented by existing systems. This distinction has received little attention in prior work, and suggests the exploration of interactions that are more heavily contextualized by slide content.

1.2 The Presentation Maestro

This thesis critically examines gesture-based input in the context of electronic presentations. To explore the aforementioned open issues, we designed, implemented, and evaluated Maestro, a prototype gesture-based presentation system. In keeping with similar systems in the literature, Maestro augments a presenter's "natural" repertoire of gestures with gestures used to navigate a presentation (e.g., next/previous slide). However, Maestro also allows users to interact *directly* with the *content* of their slideshows using gestures similar to those that have been observed spontaneously occurring during PowerPoint presentations. Some of these interactions include having bullet points automatically highlight in response to deictic (pointing) gestures; allowing presenters to zoom into figures within slides; allowing presenters to expand and collapse individual elements within a hierarchal bullet list; and allowing presenters to follow hyperlinks embedded in text.

An important aspect of Maestro's design is that hand gestures are detected using computer vision. Consequently, the system requires only the addition of a low-cost web camera and a pair of color-contrasted gloves to enable gesture-based interaction. This all but ensures that the technology can be made widely available.

Moreover, portable data projectors and powerful laptop computers (incorporating built-in web cameras) result in presentation equipment which is highly portable; this allows presenters to carry their equipment from venue to venue, and can be contrasted with related technologies such as large-scale touch sensitive surfaces. The latter are high-cost, less-widely deployed, and far less portable. The advantages of a vision-based system are very important for user acceptability, since presenters are less likely to learn the nuances of a gesture-based presentation system if they are frequently faced with presenting in venues where the technology is unavailable [39].

While the ubiquity of low-cost web cameras makes a computer vision-based approach attractive, relying on vision alone imposes many design challenges. Most importantly, a vision-based approach significantly complicates the recognition of gestures. Cameras stream observations which include any and all hand motion. A gestural interface must be able to “spot” meaningful gestures in these longer motion sequences. This issue is known as gesture segmentation, or gesture spotting [28], and is similar to keyword spotting in speech recognition. In keyword spotting, a system must be able to detect the utterance of a keyword or phrase amidst unconstrained speech and non-speech background noise. We note that issues related to gesture spotting do not arise when gestures are performed on touch sensitive surfaces, since the onset and release of contact with the surface clearly identifies which motions are intended as gestures.

1.3 Primary Contributions

Importantly, Maestro was evaluated during a two week classroom deployment study, in which one of the research supervisors used the system for lecturing to a class of approximately 100 undergraduate students. These lectures were observed by the author of this document, who attended the lectures as an audience member. The class was also asked to provide open-ended feedback, and to complete a comprehensive questionnaire regarding the use of Maestro in the classroom. To the best of our knowledge, this study constitutes the first real-world, long-term evaluation of such a system. It also includes audience feedback to an extent never before realized in previous related research.

The results of Maestro’s evaluation suggest that gestural input can have a positive impact on presentations; in particular, our results add support to the notion that gesture-based interaction leads to interactive and engaging presentations – a finding first reported by Cao *et al.* in [5]. However, our findings also suggest that gesture-based interaction can reduce the perceived efficiency of a presentation, and can noticeably alter the dynamics of a presentation in ways that are not always desirable. In particular, the requirement that the presentation be entirely controlled through gestures tends to *anchor* the presenter near the projection screen, thereby limiting his or her mobility. Additionally, the interface encourages the presenter to face the projection screen rather than the audience, thereby leading the presenter to miss audience questions and feedback. The presenter's close proximity to the screen also requires the speaker to noticeably back away from the screen to view slides in their entirety; this interrupts the presentation, and gives the impression that the presenter is unprepared. Finally, when gestures are sensed using computer vision, the presenter must be cautious when entering the volume of space directly in front of the screen; lest he risk gestures being falsely detected. We refer to this area of space as the "no-fly zone".

In addition to the aforementioned findings, Maestro's evaluation suggests that certain classes of gestures may be more beneficial than others: As noted above, Maestro makes use of two classes of gestures; those that enable *navigation* of the presentation (e.g., moving between slides), and those that allow presenters to interact with slide *content* (e.g., enabling content to respond to pointing gestures). While past research has focused on using gestures for navigation, our findings suggest that *content-centric* gestures are well received by both presenters and audience members alike; but, the benefits of *navigation* gestures are less clear. These findings were unexpected and have not been previously reported in the literature.

1.4 Secondary Contributions

Maestro was developed as a research tool with the expressed purpose of quickly and inexpensively exploring the implications of gesture-based interactions with presentations. The original gesture spotting approach used by Maestro (including in the deployment study) was heuristic in nature, and relied on manually generated gesture templates. While crude, this "ad-hoc" approach supported rapid prototyping, allowing us to *quickly* explore a heterogeneous space of gestures when designing Maestro's gesture language. For example, we experimented with both one-handed and bimanual gestures. While the ad-hoc recognizer functions quite well in practice, it is not easily generalized to cope with new gestures. In fact, adding new gestures (or improving the recognition of existing gestures) requires new heuristics to be developed on a case-by-case basis. This inflexibility lead us to develop a more principled gesture recognizer upon finalizing Maestro's design.

While Maestro's more principled recognizer uses standard discrete hidden Markov models (DHMMs) to represent gestures, our contributions lie in developing a feature set that allows modeling of both one and two-handed gestures, and which directly models missing observations (e.g. cases where noise or occlusion prevents the hands from being detected). Modeling missing observations is important not only for reasons of robustness, but also because some gestures are more likely to result in missing data (e.g. occlusions) than others; this extra information is useful in guiding gesture recognition. Moreover, these features are carefully engineered so that the DHMM's observation distributions can be easily factored using conditional independence; this greatly reduces the number of parameters that need to be learned by the system.

Since the DHMM-based recognizer was developed only after verifying Maestro's design in the classroom evaluation, we present results from a controlled experiment which directly compares the original ad-hoc recognizer to the DHMM-based recognizer. The results of this experiment reveal that both recognizers have similar recognition characteristics. This suggests that the results of Maestro's classroom evaluation are applicable to a version of Maestro which uses a gesture recognizer that is more representative to the state-of-the-art in this field.

1.5 Outline

This document is divided into two distinct parts mirroring the two distinct contributions claimed above. Part I of this thesis begins with a literature review of gesture-based presentation systems. We then present an observational study of people giving talks. This observational study motivates many aspects of Maestro's design, which we describe in Chapter 4. Following the description of Maestro's design, we present the results of Maestro's classroom evaluation.

In Part II of this document, we present a more principled gesture recognizer, as described above. Part II begins by reviewing discrete hidden Markov models, followed by a description of the DHMMs used by Maestro. In doing so, we present a feature set and an observation model which captures many important aspects of Maestro's gestures while having few parameters. We then describe the process of spotting gestures using DHMMs. Part II concludes by comparing the recognition results of the DHMM-based approach to those of Maestro's original ad-hoc gesture recognizer.

Finally, the entire thesis concludes in Chapter 9, with a summary of the findings and a discussion of future research possibilities.

Part I

The Presentation Maestro

Chapter 2

Gestural Interfaces for Presentation Software

In this chapter, we review research into the use of hand-gestures for controlling presentations. Here, the seminal research was conducted by Baudel and Beaudouin-Lafon who outlined many of the unique challenges in enabling gesture-based interaction in a presentation context. This important research is described in great detail in Section 2.1. We then review more recent research by Cao *et al.* in Section 2.2, and conclude with a survey of computer vision-based approaches which enable presenters to use hand gestures to interact with slideshows.

2.1 Charade

The Charade system, developed by Thomas Baudel and Michel Beaudouin-Lafon, is one of the earliest examples of a presentation system controlled by hand gestures [2]. This system allowed presenters to control a HyperCard presentation using a DataGlove, (also known as a "wired glove"). A DataGlove is a glove that has been instrumented with sensors which relay the joint-angles for each of the hand's finger joints. When coupled with a Polhemus tracker, the hand's location in 3D space is also measurable. Using this technology, Charade allows presenters to linearly navigate a presentation, access a table of contents, and highlight or annotate areas of each slide with free-hand drawings. To recognize gestures, Charade uses a modified Rubine recognizer, which achieves an accuracy of 70% with new users, and 90-98% with expert users. [1]

In describing the gestures used by Charade, Baudel and Beaudouin-Lafon discuss the problem of segmenting motion into discrete gestures. In regards to gesture segmentation, DataGloves suffer from the same limitations as vision-based approaches; these systems record all hand motion, not just the motion that is intended as a gesture. In order to resolve this challenge, the Charade system

[1] These error rates are similar to those of Maestro, as reported in Chapter 8.

requires that presenters begin and end each gesture with the hands in a "tense" posture, such as a clenched fist, or an open hand with the fingers stretched widely apart. These tense hand postures are said to be "non-usual, but not unnatural". Additionally, to avoid user fatigue, the gestures are designed so that they can be performed quickly without requiring much accuracy on the part of the users. Gestures that are most frequently used, such as next and previous slide, are assigned the quickest, most natural gestures. To limit recognition errors, CHARADE ignores hand motion that does not occur within an "active space" directly in front of the screen.

Baudel and Beaudouin-Lafon also discuss the challenge of providing affordances and feedback in a gesture-based system. They note that "good feedback is mandatory because gestural commands are not self-revealing". The authors suggest that three types of feedback be provided: syntactic feedback, which reveals the state of the gesture recognizer [2]; semantic feedback, which conveys the effect of a gesture command; and command history feedback which reveals the past sequence of commands. The command history allows users to recover from cases where gestures are falsely detected. In this last case, a general "undo" command is essential.

In discussing both the challenge of gesture spotting and the challenge of providing user feedback, Baudel and Beaudouin-Lafon's work on CHARADE is perhaps the most thorough description of a gesture-based presentation system in the literature. Nonetheless, this research left many issues unresolved. Some of these open problems are described below:

- REAL-WORLD EVALUATION
  CHARADE was reportedly used by two trained users to present sample presentations to an audience. The authors described the purpose of the trial as "to determine whether the application was useable in a real setting". However, the only results reported from this trial were the error rates, along with the observation that most errors were noticed immediately (presumably by the presenter). Additionally, there was no mention of the audience's size or of the duration of these presentations. A far more thorough real-world evaluation is thus warranted.

- THE AUDIENCE
  There are two distinct "users" who benefit from presentation software: the presenter, and the audience. The usability of a presentation system depends both on the presenter's ability to effectively command the presentation, and the audience's ability to effectively interpret or comprehend the presentation (e.g., without being distracted by the interface). The literature describing CHARADE largely ignores the audience. For example, Baudel and Beaudouin-Lafon describe the need to provide feedback to the presenter, but they do not

[2] The authors suggest that syntactic feedback be communicated by changing the shape of the cursors. This is the strategy that Maestro uses.

mention if the audience is distracted by this feedback. Similarly, the authors mention that the feedback allows presenters to recognize and correct errors in “one or two gestures”, but they do not mention how these errors affect the audience’s perception of the presentation.

- Interacting with content
  In describing the advantages of gesture-based interaction, Baudel and Beaudouin-Lafon mention that such interfaces allow direct manipulation. However, the gestures used by Charade are largely context-independent; the gestures manipulate the presentation slide deck (e.g., changing slides), not the contents of the individual slides. We find gesture-enabled direct manipulation to be a compelling possibility worthy of additional research.

2.2 A Comparison of Presentation Control Modalities

While Baudel and Beaudouin-Lafon’s work on Charade focused primarily on the needs of the presenter, a study conducted by Cao *et al.* explored how audiences respond to gesture-based control of a presentation [5]. In this Wizard of Oz study, six individuals were asked to present talks in front of test audiences. For each audience, the presenters were asked to control the presentation using either a standard keyboard and mouse, a laser pointer equipped with a button [3], or hand gestures and a touch-sensitive screen. Since this was a Wizard of Oz study, neither the bare-handed interaction, nor the laser-pointer system were actually implemented. Instead, one of the experimenters controlled the presentation in response to actions performed by the presenters. After each presentation, audience members were asked to rate the presentation for clearness, efficiency and attractiveness using a 7-point Likert scale. Hand gesture interaction consistently received the highest score in all categories, beating the laser pointer and the keyboard by a wide margin: 70% of the audience and 83% of presenters stated that they preferred the use of hand gestures.

Despite the positive results from the audience questionnaire, the researchers found both advantages and disadvantages to gesture-based interaction. Specifically, this input modality was found to be natural and easy to use, and was found to encourage the increased use of body language; however, presenters tended to remain near the screen rather than walk around, and presenters occasionally occluded the audience’s view of the screen. In addition to these findings, the researchers reported that gesture-based interaction requires an easy “undo” operation, and a fast “next

[3] Laser pointer systems use computer vision techniques to track the laser dot on the projection screen in order to direct a cursor. Such systems have been explored by numerous researchers [55, 54, 42, 6, 25]

slide" command. These latter recommendations mirror those made by Baudel and Beaudouin-Lafon in their work on Charade.

While Cao's study provides evidence for the benefits of gesture-based interaction with presentation systems, a number of important research questions remain. For example, it is unclear if (or how) the Wizard of Oz study simulated recognition errors which are bound to occur – even if only rarely. These details are important if we are to understand how audiences respond to such errors. The study also presumed a touch-based interface, while many gesture-based systems rely on computer vision techniques (see Section 2.3). It is thus unclear how use of computer vision affects presentation dynamics, especially since it is more likely to generate recognition errors. Most importantly, the study was also limited in scale: gesture-based interaction was evaluated for a total of six talks, each of which was only five minutes in duration. As such, it is unknown how well these systems fare in more regular, day-to-day use.

2.3 Single-camera vision-based recognition of hand gestures for controlling presentations

In the previous sections we described Charade, which relied on a DataGlove for sensing hand gestures; and Cao's observational study, which assumed the use of a large touch-sensitive display. In both cases, the authors suggested computer vision as an alternative method for sensing gestures. In particular, Baudel and Beaudouin-Lafon complained that the DataGlove was a major limiting factor in the Charade system, since the DataGlove was too large to properly fit the hands of two participants of their system's small user study [2].

More than 15 years have passed since Charade was first developed. Inexpensive web cameras are now ubiquitous, and modern computers have sufficient processing capabilities to make computer vision a viable alternative for detecting hand gestures. Vision-based approaches, employing the use of a single camera, are advantageous on account of their wide availability, low cost, and high portability. These benefits should not be undervalued; in evaluating the benefits of interactive whiteboards (a type of large touch sensitive-displays) in schools, Levy reports that a teacher's proficiency and creativity in using the display depends on "easy and frequent" access to the technology in order to support experimentation [29]. Similarly, Smith *et al.* report that "there is little incentive for secondary school teachers to plan a lesson with the (interactive whiteboard) if they are faced with repeating the same lesson in a room without the board" [52]. The low cost and portability of a vision-based approach ensures that presenters can acquire and easily transport their presentation equipment from venue to venue.

While the ubiquity of low-cost web cameras makes computer vision-based approaches attractive, such systems are often challenging to construct because they must be able to spot gestures embedded within sequences of more general hand motion. This is especially challenging in a presentation environment because presenters often use a great deal of gesticulation, but only occasionally issue gestural commands. The literature describes a number of computer vision-based gestural presentation systems, each of which can be categorized according to the approach used for gesture segmentation. We describe each in turn.

2.3.1 Gesture segmentation by "dwelling"

Perhaps the simplest approach to segmenting gestures is to require that all gestures begin or end with a sustained period of zero velocity. This strategy is especially common with deictic gestures, where commands can be issued by pointing to a target for some length of time. This gesture is known as *dwell clicking* and can be used to emulate a one-button mouse. The approach is commonly utilized in gaze interfaces which allow users to control a computer using eye movement [17]. There are numerous presentation systems where dwell clicking gestures are used. For example, Sukthankar *et al.* developed a system where users can navigate slides, and annotate slides with freehand drawings [55] by pointing to active regions on the screen with an outstretched finger. To activate a region, users must point to the target for approximately half of a second. A similar presentation system was also described by Noi Sukaviriya *et al.* in [53].

The *FreeHandPresent* system by Von Hardenberg and Bernard demonstrates how dwell clicking can be augmented to differentiate between several hand postures [59]. Overall, their work focused on the problem of detecting hands in cluttered images, and did not focus much on the design of a presentation system. *FreeHandPresent*'s interface is very limited in that only three commands are available to the presenter: extending two fingers instructs the presentation to advance to the next slide; three outstretched fingers signals the system to return to the previous slide; and five outstretched fingers opens a "slide menu" which allows presenters random access to any slide. As with simple dwell clicking, gestures were signaled by holding the hand still, with a particular posture, for an unspecified length of time (presumably between 0.5 seconds and 1 second). A similar system was described by Licsar and Sziranyi in [30], who note that more complex gestures can be synthesized by requiring that users perform a sequence of static hand postures in order to step through a finite state machine (FSM). In this sense, a gesture is any sequence of static hand postures that satisfies the grammar defined by the FSM.

Each of the aforementioned systems uses dwelling to initiate the recognition of a static hand posture. It is also possible to use dwelling to initiate the recognition of a dynamic gesture, in which the gesture is performed by moving the hand along some path through space. This is the approach taken by Yoon *et al.* in [68], where HMMs are used to recognize dynamic gestures whose start and endpoints are indicated by dwelling. While they did not incorporate their approach into an electronic presentation system, their work is certainly worth mentioning.

While it is clear that "dwelling" is an effective means for gesture segmentation, it suffers one major drawback: it is difficult to set a satisfactory minimum duration for the dwelling. If the dwell duration is set too high, then the system feels less responsive, and users can become impatient. When experimenting with dwelling in Maestro, we noticed that it is difficult to hold the hand still while talking, as this goes against the natural tendency to gesticulate. Consequently, users often stop talking while waiting for the system to respond to the dwell click gestures. This noticably interrupts the flow of the presentation. Unfortunately, responding to this problem by lowering the dwell duration leads to the "Midas touch" problem [17], where gestures may inadvertently be activated whenever, and wherever, the hands rest. It is quite difficult to achieve an appropriate balance.

2.3.2 Gesture segmentation by electronic signaling

Another common approach to addressing the gesture segmentation problem is to require the user to depress a wireless button, or to otherwise electronically signal, the onset or termination of every gesture. There are several commercial products that use this strategy for gesture segmentation. GestureStorm, produced by Cybernet Systems, allows television network meteorologists to interact with their weather maps "through a combination of hand motions in front of a green or blue screen and button clicks on a wireless remote" [9]. For example, circling an area of the map might cause the system to zoom into a particular region or district. GestureStorm resolves the gesture segmentation problem by having presenters click a button on the remote at the onset of each gesture. This system relies on a green screen to detect and track the presenter, which makes it unsuitable for everyday presentations.

Additionally, iMatte Incorporated briefly produced a product called iSkia, a small appliance that attaches to a standard LCD projector. This appliance floods the projection screen with infrared light, and uses an infrared camera to detect the presenter's contour. The system directs a mouse cursor by finding the point on the contour which is furthest from the contour's horizontal center. If the presenter's hand is outstretched, this point corresponds to the presenter's fingertip. Otherwise, it might correspond to the presenter's shoulder, elbow, or other extremity. A wireless remote then allows presenters to issue mouse click events. While not a gesture-based interface per se (since the hand emulates a mouse), it nonetheless provides more direct interaction than using a wired mouse or keyboard.

Electronic signaling is certainly a viable option for segmenting gestures, however the resulting solutions require additional equipment (e.g: a wireless remote), and thus no longer rely exclusively only on computer vision. Moreover, the addition of a remote reintroduces a physical transducer mediating the interaction between the presenter and the presentation software; the presenter must now interact with the physical remote in order to manipulate their presentation.

2.3.3 Gesture segmentation in continuous motion

Finally, there exist numerous strategies [28, 24, 3] for segmenting dynamic gestures in sequences of motion (i.e., without requiring dwelling or electronic signaling). We describe these approaches in Part II of this document, focusing now only on techniques that have been incorporated into gesture-based presentation systems. Along these lines, only one such system was uncovered in our survey. Lee and Kim's PowerGesture [28] system provides a gesture-based front-end to Microsoft PowerPoint. In this system, ten separate gestures can be recognized in streams of continuous hand motion. Gesture recognition is driven by a network of discrete hidden Markov models (DHMMs). To address the gesture segmentation problem, Lee uses a "threshold model" which gives a time-varying likelihood threshold that can be used in conjunction with the gesture DHMMs for gesture spotting. With PowerGesture, users can manipulate the presentation (e.g., navigate back and forth in the slides, or quit the presentation), but cannot interact with individual elements on the slides.

2.4 Discussion

Gesture-based interaction with electronic presentation systems has a long history in the literature. The earliest system, Charade, was initially described more than 15 years ago. Nevertheless, there remain some very large open issues which need to be addressed before gesture-based presentation control can be considered a viable alternative to keyboards, mice and wireless remotes. First, there is a systemic lack of real-world evaluation data to assess the usefulness of gesture-based presentation systems. While authors often report anecdotes of using their software to give presentations [55, 2], Cao's Wizard of Oz study remains the only body of work soliciting feedback from both the audience and the presenter. Cao's study is also the only body of research to use a standardized survey to enable quantitative statistical analysis regarding the useability of a gesture-based presentation system. Unfortunately, Cao's results are limited in scale, analyzing only a total of 30 minutes of data.

The second open issue regards the richness of the gesture-based interactions. Gesture-based interaction promises to enable rich, direct interaction with the contents of every projected slide. However, the systems described in the literature typically only enable simple commands such as "next slide" and "previous slide".

Subsequent chapters of part I of this document present the design and evaluation of a gesture-based system which addresses the aforementioned open issues. Importantly, the design of this system is motivated by an observational study of people giving talks. This observational study is the subject of the next chapter. Later, in part II of this document, we revisit the problem of reliably segmenting (i.e., spotting) gestures embedded in continuous hand motion.

Chapter 3

Observational Study

In this chapter we describe a small observational study of people giving talks. This study was conducted in order to observe the hand gestures that naturally arise throughout the course of a presentation. Our observations directly informed the design of Maestro. In presenting our observations we review vocabulary for describing and classifying gestures. This vocabulary will be used throughout this document.

3.1 Introduction

In the previous chapter, existing gesture-based presentations systems were reviewed. These past systems were often motivated by the observation that individuals tend to gesture when giving a presentation; unfortunately, how presenters gesture, under what circumstances, and for what purposes, is not generally discussed. This information is critical to help create designs that naturally integrate with current presentation practices. While not specific to gesture-based interfaces to presentations, there is some literature which begins to address these issues. Most notably, Joel Lanir *et al.* conducted an observational study of people giving presentations in order to determine how best to leverage multiple data projectors in large lecture halls. While this study did not focus on gestures, it provides a good foundation for understanding the contexts in which gestures are most-likely to arise. The study reported that presenters gesture to approximately 17% of slides containing only text, increasing to 88% when slides contain figures or tables. They also noted that gestures tend to draw the audience's focus, and are used to "connect the audio and visual parts of a presentation". Unfortunately, few details were provided regarding the specific gestures that were observed.

Other related research includes work done by Shanon Ju *et al.* , who categorized hand motions that arise during technical talks presented using an overhead projector [20]. Here, three classes of gestures were identified: pointing using a pen or one's finger, writing on the transparencies, and incrementally revealing text by removing a piece of paper from the transparency. This work was used to inform the design of a system which automatically annotates videos of technical talks so that they can be indexed by information retrieval systems.

More generally, the description and categorization of natural speech-associated gestures has been widely studied in psycholinguistics, which studies the cognitive processes involved with language [13]. We briefly review this literature in Section 3.2 in order to introduce the concepts and vocabulary needed to better describe gestures. We then present our observations and design recommendations which we derived from a small observational study of people giving presentations.

3.2 Gesture taxonomy

In the field of psycholinguistics there has been a great deal of interest in describing and categorizing gestures that arise from conversational speech. While a detailed review of this literature is beyond the scope of this document, interested readers are referred to the University of Chicago's McNeill lab for Gesture and Speech Research [35]. In reviewing this literature, one fact emerges: there is no clear consensus on how to describe gestures [44, 61]. For example, in categorizing gestures there are at least four competing gesture taxonomies [61], each using its own terminology. Additionally, researchers often combine ideas from these taxonomies [44, 23], a practice which is complicated by the fact that the taxonomies are often incompatible. As pointed out by Alan Wexelblat there are even cases of competing taxonomies using the same term to mean two different things [61]. For this reason, we spend the next section of this document carefully defining our choice of terminology.

3.2.1 Gesture form vs. Gesture function

The description of gestures can be divided along two orthogonal axes: the first describes the gesture's function (manipulative vs. communicative, etc.), the second describes the gesture's form (one-handed vs. bimanual, etc.). For example, one can describe a particular sequence of hand motion as a "bimanual manipulative gesture" or a "one-handed communicative gesture". In this chapter the focus is on formally categorizing gestures by their function. We take a less formal approach to describing gesture form.

3.2.2 Gesture function

In categorizing a gesture's function, we are interested in assigning the gesture to a single category within a traditional gesture taxonomy. The taxonomy that we have elected to use is a slightly modified version of the taxonomy described by Pavlovic *et al.* in [44], which is in turn based on writings by Francis Quek, David McNeill, and others [46, 34]. Our version of the taxonomy (figure 3.1) favors simplicity over granularity, yet it captures many aspects of gesture that are important for interface designers. Although the taxonomy is presented as a tree, the taxonomy's categories consist only of the tree's leaves: manipulative, beat, deictic, ideographic, and semaphoric gestures. The inner nodes simply label semantically meaningful groupings of the categories. For example, while "gesticulation" tends to be highly associated with speech, the grouping is simply defined as the set of gestures that can be classified as either beat gestures, deictic gestures or ideographic gestures. We describe each of the five categories below, which are arranged from least language-like to most language-like from left to right.

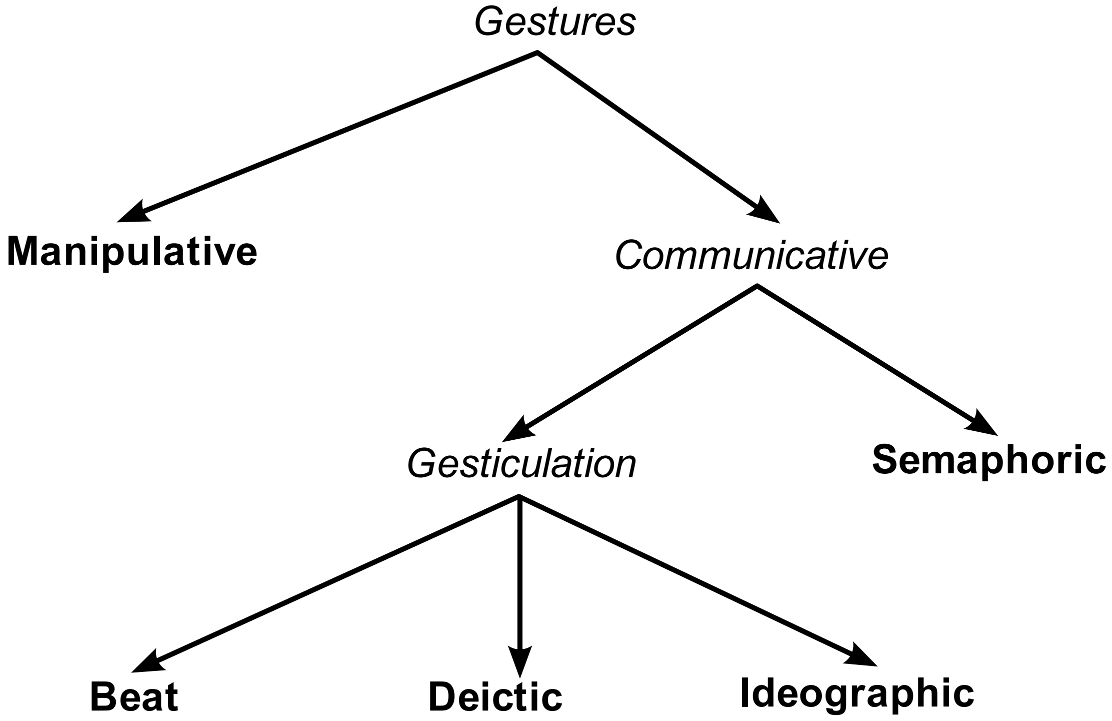


Figure 3.1: A taxonomy of gesture functions, organized from least language-like to most language-like (from left to right).

Manipulative gestures

Francis Quek defines manipulative gestures as

> those (gestures) whose intended purpose is to control some entity by applying a tight relationship between the actual movements of the gesturing hand/arm with the entity being manipulated. [46]

An example is the "drag and drop" mouse gesture, which is used in many direct manipulation-style interfaces. Importantly, Manipulative gestures are characterized by a tight "perception-action" loop, often requiring a great deal of hand-eye coordination, or tactile feedback.

Beat gestures

David McNeill describes beat gestures as

> among the least elaborate of gestures formally. They are mere flicks of the hand(s) up and down, back and forth that seem to 'beat' along with the rhythm of speech. [34]

Beat gestures tend to emphasize portions of the discourse which the speaker finds important.

Deictic gestures

Deictic gestures are simply pointing gestures, which serve to specify an entity (person, place, or thing), or a group of entities. Here, the gesture's information content is almost completely derived from the spatial context in which the gesture occurs.

Ideographic gestures

Ideographic gestures are those that depict the ideas or concepts expressed in speech. This category combines McNeill's Iconic gesture category (where gestures depict the literal content of speech[34]), and McNeill's Metaphoric gesture category (where gestures depict the speaker's ideas, but not the literal speech content). Examples of such gestures include holding one's hands far apart when discussing quantities that are large, tracing the shape or contour of an object in the air, or holding two hands outward as if to weigh two competing concepts (similar to the spoken idiom "on the one hand ... on the other hand").

Semaphoric gestures

Semaphoric gestures are those whose form and function are related through a well-defined gesture dictionary (American Sign Language [57], US Army Field Manual FM 21-60: Visual Signals [1], etc.), or through established cultural norms ("thumbs up", peace, etc.). Importantly, semaphoric gestures express complete commands or ideas, and often replace spoken language (especially in environments where spoken language may be inaudible).

Having now established a vocabulary for categorizing and describing hand gestures, we proceed with the description of our observational study of people giving talks.

3.3 Observational Study

In order to conduct our observational study of people giving talks, we selected 8 videos from Google's Tech Talks website. These videos consisted of 8 individuals lecturing for a total of approximately $7\frac{1}{2}$ hours. In each instance, the lecturers presented an electronic presentation which was front-projected onto a small projection screen typical of a classroom or boardroom. Importantly, most areas of the screen could be accessed by the presenters, although they may have had to walk around to reach the far-left or far-right sides.

Immediately obvious from these videos was that presenters performed a great many gestures, all of which can be categorized as gesticulation according to the aforementioned taxonomy. This was not surprising since these spontaneous speech-associated gestures are one of the most diverse forms of non-verbal communication, representing more than 90% of all human gestures [33]. Additionally, we noted that the presence of visual aids (the projected slideshow) vastly altered the dynamics of the gesticulation, as compared to conversational speech. For example, the aforementioned gesture taxonomy describes beat gestures as hand "flicks" that follow the rhythm of the speech and which allow the speaker to emphasize certain words. When lecturers are speaking in the context of an electronic presentation, they still perform many beat gestures, but also use a great many deictic gestures to emphasize words. This is done by actually pointing to the written words displayed within the slides - something that is not possible in conversational speech. Perhaps even more interesting is that, when pointing to words on the screen (a deictic gesture), the presenters continue to move their hands to match the rhythm of speech (a property of beat gestures). In this sense, these rhythmic deictic gestures are formatively different than standard deictic gestures where one is pointing to a person, place, or object.

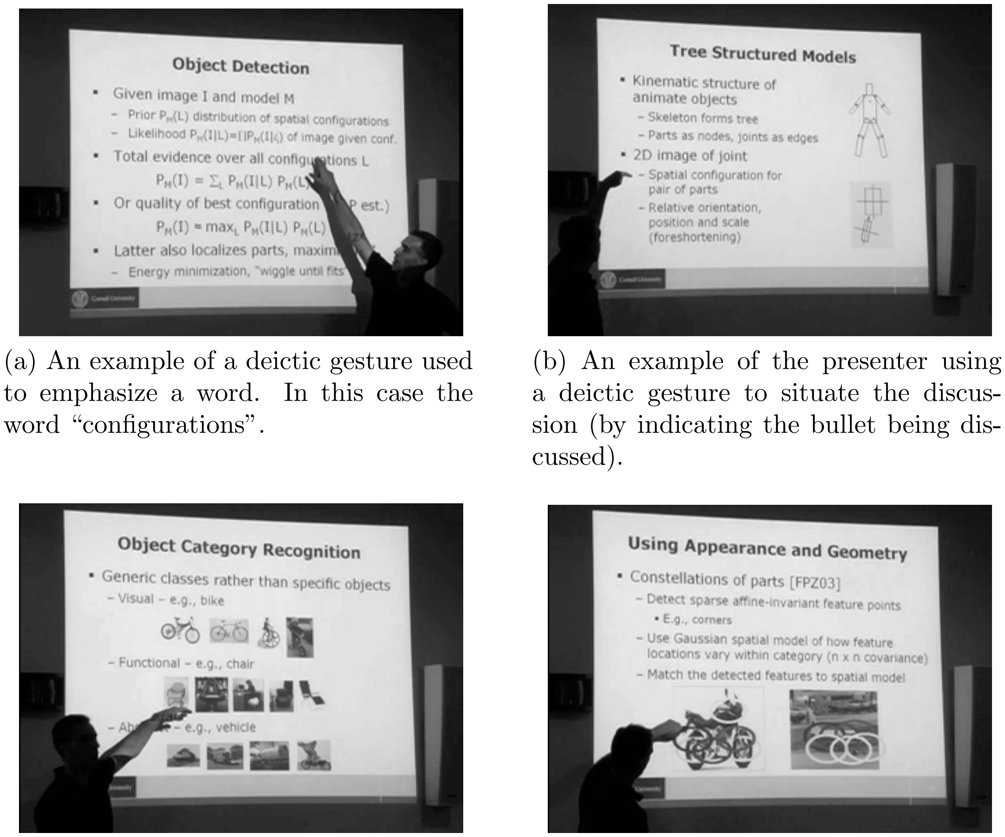
Since the projection screen noticeably alters gesticulation, and because the observational study was conducted in order to inform the design of a presentation system, we focused our attention on those gestures which were performed in front of, or near, the projection screen. We now describe the gestures whose form and apparent function were consistent across numerous presentations and presenters.

Emphasizing words using deictic gestures

As already mentioned, presenters often pointed, underlined, or circled words in slides in order to emphasize those words while speaking (as in figure 3.2a).

Situating the discussion using deictic gestures

In addition to emphasizing words using deictic gestures, presenters often pointed to entire bullet points or phrases (as in figure 3.2b). In these cases, presenters pointed to the left or right extremities of the bullet point, with the bullet graphic



Object Detection

- Given image I and model M
  - Prior $P_M(L)$ distribution of spatial configurations
  - Likelihood $P_M(I|L)=\prod P_M(I|l_i)$ of image given conf.
- Total evidence over all configurations L

$$P_M(I) = \sum_L P_M(I|L)\ P_M(L)$$

- Or quality of best configuration P est.)

$$P_M(I) \approx \max_L P_M(I|L)\ P_M(L)$$

- Latter also localizes parts, maxim
  - Energy minimization, "wiggle until fits"

(a) An example of a deictic gesture used to emphasize a word. In this case the word "configurations".

Tree Structured Models

- Kinematic structure of animate objects
  - Skeleton forms tree
  - Parts as nodes, joints as edges
- 2D image of joint
  - Spatial configuration for pair of parts
  - Relative orientation, position and scale (foreshortening)

(b) An example of the presenter using a deictic gesture to situate the discussion (by indicating the bullet being discussed).

Object Category Recognition

- Generic classes rather than specific objects
  - Visual – e.g., bike
  - Functional – e.g., chair
  - Ab – e.g., vehicle

(c) An example of the enumeration gesture. The presenter continues this gesture by pointing to the four other images along the row.

Using Appearance and Geometry

- Constellations of parts [FPZ03]
  - Detect sparse affine-invariant feature points
    - E.g., corners
  - Use Gaussian spatial model of how feature locations vary within category (n x n covariance)
  - Match the detected features to spatial model

(d) An example of a tracing gesture. The presenter performs this gesture by tracing the circle depicted on the slide.

Figure 3.2: Various deictic and iconic gestures observed in the Tech Talks videos.

often serving as a natural target for these gestures. Rather than serving as a form of visual emphasis, these gestures helped situate the discussion by indicating which subtopic was being discussed. Moreover, if the bullets were being discussed in sequential order, this deictic gesture also indicates the progression through the slide.

Listing or enumerating items

Another deictic gesture, presenters were often observed pointing to numerous items in rapid succession (as in figure 3.2c). This gesture was used to group objects or to indicate membership in a set. When referring to the entire set, presenters often waved their hands over all items, without indicating any one item in particular.

Tracing shapes or lines

Presenters were often observed tracing lines in a line graph, following arrows in a flowchart, or tracing the contours of objects in images (as in figure 3.2d). In some cases, this gesture served to "animate" a still image; for example, to indicate the trajectory of a moving object. The gesture taxonomy lists such shape-tracing gestures as ideographic gestures that help visualize the content of speech. In presentations, much of the contents of speech are already visualized through figures or images. Nonetheless, presenters continue to trace shapes while speaking - but now the gestures are performed against an image on the projection screen rather than in the air.

Cropping figures

In at least four cases, we observed presenters "cropping" or "framing" portions of an image or figure (depicted in figure 3.3). This is a deictic gesture, serving to isolate a region of an image from the rest of the figure with the purpose of indicating which aspect of the figure the presenter is currently discussing. This gesture was especially interesting in that it employed the use of two hands.

Other regularities

In addition to these recurrent gestures, we also noted various other tendencies regarding the use of gestures while presenting. First, presenters typically gestured from a position just outside the left or the right edge of the projection screen - rarely standing in front of the projected display. These findings echo those of Cao *et al.* [5], who found a natural tendency for presenters to orient themselves next to the projection screen. There are two possible explanations for this behaviour. First, standing directly in front of the projection screen would occlude portions of the slide from the audience's view. This behavior is clearly undesirable, and should

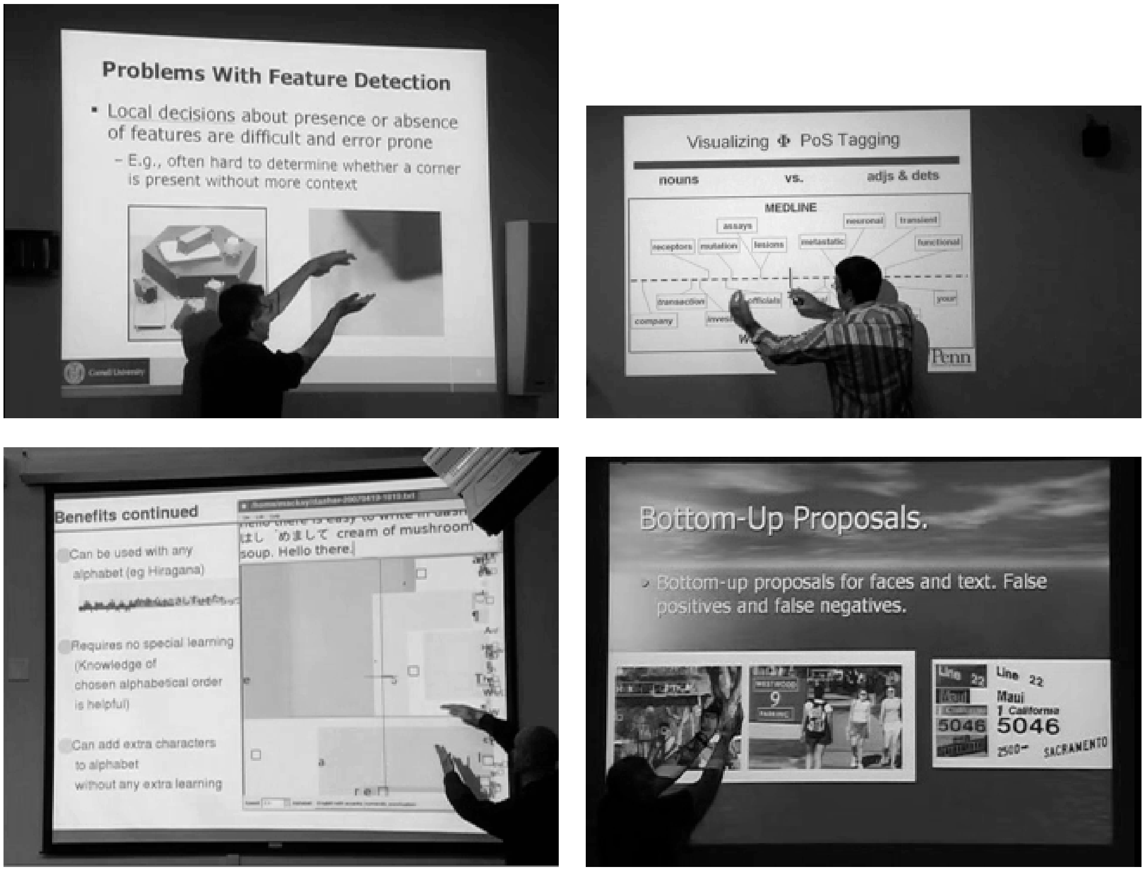
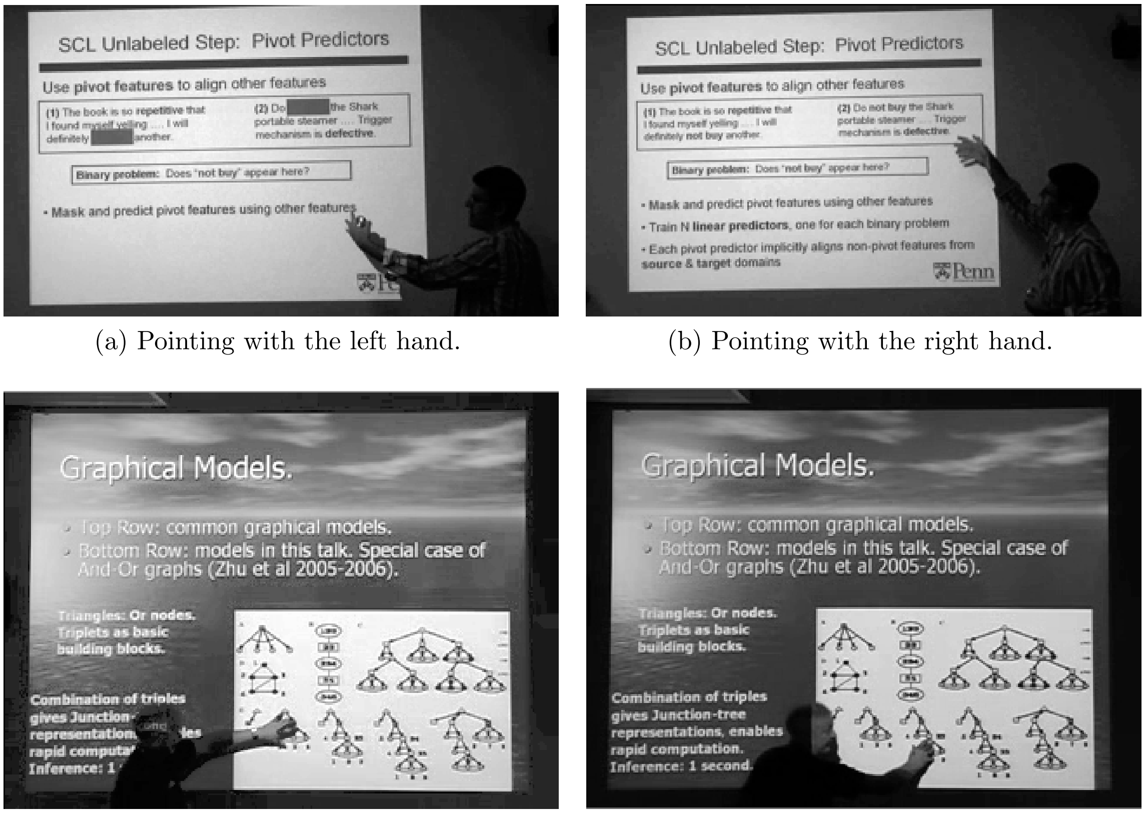

Problems With Feature Detection
Local decisions about presence or absence of features are difficult and error prone
– E.g., often hard to determine whether a corner is present without more context
Visualizing Φ PoS Tagging
nouns
vs.
adjs & dets
MEDLINE
Penn
Benefits continued
Can be used with any alphabet (eg Hiragana)
Requires no special learning (Knowledge of chosen alphabetical order is helpful)
Can add extra characters to alphabet without any extra learning
はじめまして cream of mushroom soup. Hello there.
Bottom-Up Proposals.
Bottom-up proposals for faces and text. False positives and false negatives.
Line 22
Maui
5046
SACRAMENTO

Figure 3.3: Various instances of the “cropping” gesture.

be avoided. Second, the light from the projector is very bright and can distract or disorient the presenter [51, 56, 15].

The second observation regarding the use of gestures during a presentation is that, while each presenter demonstrated a clear bias in handedness, presenters occasionally used both hands interchangeably (figure 3.4). When hand preferences were observed, they appeared to be dependent on the presenter's position relative to the screen rather than on their natural handedness. For example, when pointing, presenters used whichever hand allowed them to continue to face the audience while speaking. Consequently, they used opposite hands when standing on opposite sides of the projection screen.


SCL Unlabeled Step: Pivot Predictors
Use pivot features to align other features
(1) The book is so repetitive that I found myself yelling .... I will definitely [redacted] another.
(2) Do [redacted] the Shark portable steamer .... Trigger mechanism is defective.
Binary problem: Does "not buy" appear here?
• Mask and predict pivot features using other features
SCL Unlabeled Step: Pivot Predictors
Use pivot features to align other features
(1) The book is so repetitive that I found myself yelling .... I will definitely not buy another.
(2) Do not buy the Shark portable steamer .... Trigger mechanism is defective.
Binary problem: Does "not buy" appear here?
• Mask and predict pivot features using other features
• Train N linear predictors, one for each binary problem
• Each pivot predictor implicitly aligns non-pivot features from source & target domains
Penn
Graphical Models.
• Top Row: common graphical models.
• Bottom Row: models in this talk. Special case of And-Or graphs (Zhu et al 2005-2006).
Triangles: Or nodes.
Triplets as basic building blocks.
Combination of triples gives Junction-tree representations, enables rapid computation.
Inference: 1 second.

(a) Pointing with the left hand.

(b) Pointing with the right hand.

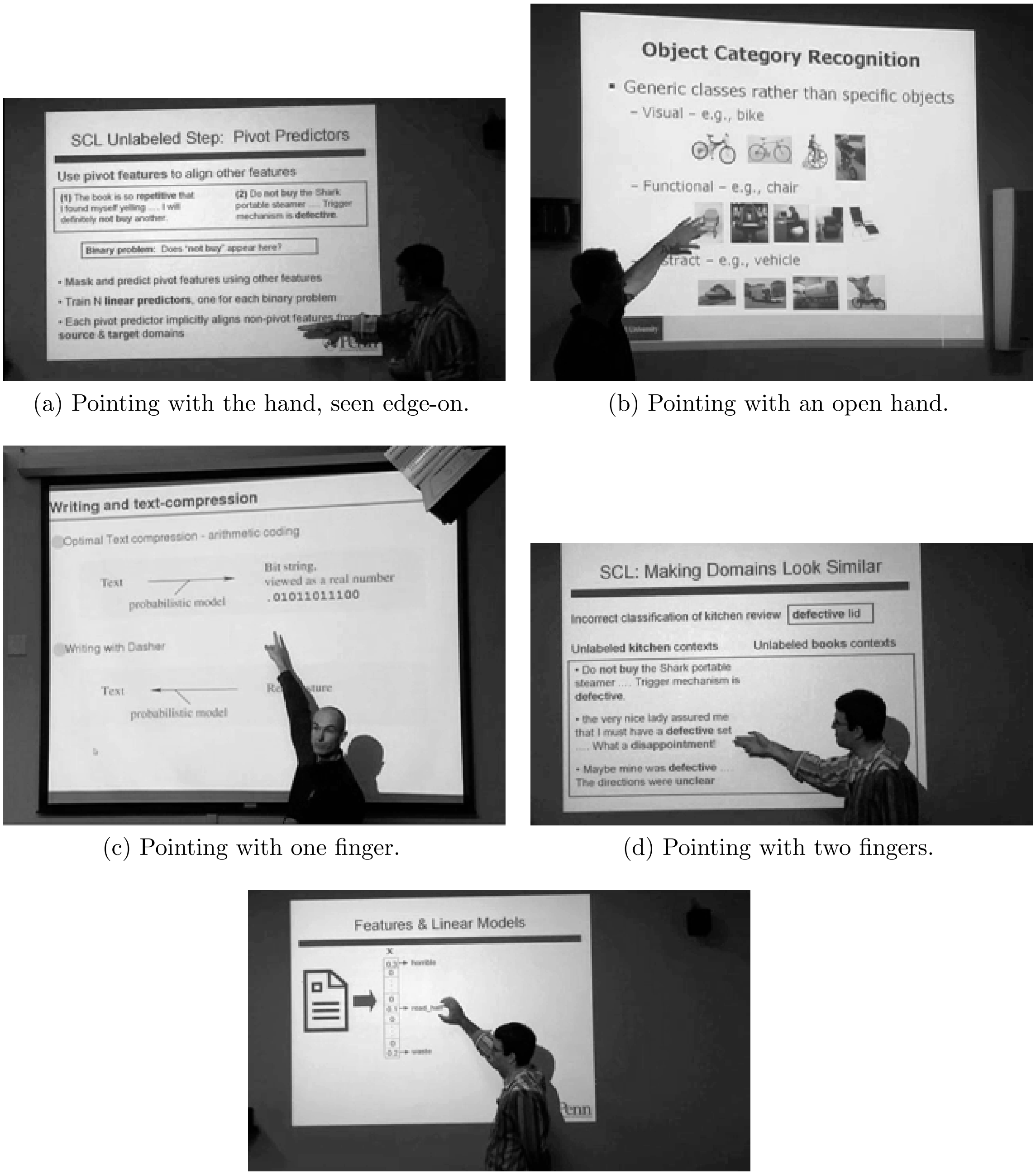
(c) Pointing with the left hand.

(d) Pointing with the right hand.

Figure 3.4: Examples of presenters using their left and right hands interchangeably.

We also noticed that presenters employed a wide variety of hand postures – even for the same gesture. For example, when pointing to a word, presenters may point with either one finger (figure 3.5c) or two fingers (figure 3.5d), an open hand (figure 3.5b), or the hand seen edge-on (figure 3.5a). As with handedness, the use of hand postures appears to be interchangeable, and does not noticeably effect the apparent meaning of the gesture.

Having described the gestures observed in the observational study, we now describe how these observations informed the design of Maestro.


SCL Unlabeled Step: Pivot Predictors
Use pivot features to align other features
(1) The book is so repetitive that I found myself yelling .... I will definitely not buy another.
(2) Do not buy the Shark portable steamer .... Trigger mechanism is defective.
Binary problem: Does "not buy" appear here?
• Mask and predict pivot features using other features
• Train N linear predictors, one for each binary problem
• Each pivot predictor implicitly aligns non-pivot features from source & target domains
Object Category Recognition
▪ Generic classes rather than specific objects
– Visual – e.g., bike
– Functional – e.g., chair
– Abstract – e.g., vehicle
Writing and text-compression
Optimal Text compression - arithmetic coding
Text
probabilistic model
Bit string, viewed as a real number
.01011011100
Writing with Dasher
Text
probabilistic model
SCL: Making Domains Look Similar
Incorrect classification of kitchen review
defective lid
Unlabeled kitchen contexts
Unlabeled books contexts
• Do not buy the Shark portable steamer .... Trigger mechanism is defective.
• the very nice lady assured me that I must have a defective set .... What a disappointment!
• Maybe mine was defective .... The directions were unclear
Features & Linear Models
x
horrible
read_half
waste
Penn

(a) Pointing with the hand, seen edge-on.

(b) Pointing with an open hand.

(c) Pointing with one finger.

(d) Pointing with two fingers.

(e) Pointing with a "cupped" hand posture.

Figure 3.5: Various hand postures used for deictic gestures.

3.4 Design implications

Prior gesture-based presentation systems have focused exclusively on using arbitrary semaphoric gestures to *navigate* a slideshow (e.g., using two or three outstretched fingers to move forward and backward between slides [59]). These gestures rarely depend on slide content, and do not resemble the types of gestures that naturally occur during a presentation. The observational study found that naturally occurring gestures are classified as gesticulation, and serve *communicative* purposes; they complement the verbal presentation by drawing the audience's attention to particular features of the visual presentation. In this sense, they are highly contextualized, and depend heavily on the content and layout of the slides. While Maestro (as with all other gesture-based presentation systems in the literature), can only recognize a limited set of semaphoric gestures, it is reasonable to design gestures that resemble natural gesticulation. This leads us to further classify Maestro's semaphoric gestures into two categories: *presentation navigation gestures* and slide *content gestures*. We describe each of these subcategories below:

- **Slide content gestures**
  - resemble gesticulation,
  - are heavily contextualized by slide content and layout,
  - are designed to supplement the verbal presentation,
  - and are directed at *both* the audience and the system.
- **Presentation navigation gestures**
  - do *not* resemble gesticulation,
  - are used to advance the slideshow,
  - are independent of slide content,
  - and are directed *only* at the system.

Where possible, Maestro's gestures are inspired by gesticulation observed in the observational study. For example, Maestro allows presenters to specify objects using deictic gestures, and uses a gesture very similar to "cropping" in order to zoom into a figure. Even if gestures do not directly resemble gesticulation, we make the following recommendations in order to design gestures which share many properties with this natural form of gesture:

- The gesture recognition system should treat each hand interchangeably, allowing gestures to be performed by either hand. In other words, the meaning of a gesture should not depend on which hand was used.

- The system should avoid the use of artificially-imposed hand postures (e.g., requiring a “thumbs up” posture) since presenters use many hand postures while presenting, and these postures tend to be interchangeable.
- The system should avoid the use of overly complex stroke gestures (e.g., stroke patterns associated with written characters) since natural gesticulation tends to consist of brief (efficient) emphatic motion.
- Gestures should be performed from a position just outside the left or the right side of the projection screen where the presenter is unlikely to occlude the audience’s view of the slide, and where the presenter is unlikely to be distracted by the projector light.

The danger in designing a gesture language based around gesticulation is that it becomes increasingly difficult to distinguish between spontaneous gesticulation and the semaphoric gestures that are indented to elicit a response from the system. The ability to ignore gesticulation while still recognizing the semaphoric gestures is crucial so as to prevent suppressing one’s natural tendency to gesticulate; if presenters adapt to recognition errors by avoiding gesticulation, then the presentation will lose an important aspect of non-verbal communication. In the next chapter, we present the design of a gesture-based presentation system which addresses this challenge, as well as other challenges related to gesture-based presentation control. This gesture-based presentation system, Maestro, is directly inspired by the results of the observational study described in this chapter.

Chapter 4

Designing Maestro

Upon completion of the observational study, work began on designing Maestro, a gesture-based presentation system that requires only a web camera for input. In this chapter, we present Maestro's design, and present some of the lessons we learned in early stages of user testing. The evaluation of Maestro's final design is presented in Chapter 5, while the evaluation of Maestro's gesture recognizer is presented later, in Chapter 8.

4.1 Design goals and challenges

Maestro is a presentation system which allows presenters to use hand gestures to both navigate a projected slideshow and to interact with the individual components of each slide. Maestro's design is guided by the following ideals:

1. Software Features and the Gesture Language
   Users should be able to navigate the presentation (e.g., move between slides), as well as interact with elements within slides (e.g., bullet points) using gestures that are similar to those observed in the observational study. However, these gestures must be designed in such a way that they are not erroneously recognized when the presenter is gesticulating.

2. Feedback, Affordances and Error Recovery
   The system should be easily learned, provide affordances for its use, offer appropriate feedback during use, and support swift recovery from recognition errors. However, achieving these overall useability goals should not interfere with the dynamics of the presentation nor with its visual appearance.

3. Hand Detection and Gesture Spotting
   Maestro should rely only on a *single* web camera for input, and a data projector for output. However, this configuration complicates the task of tracking the presenter's hands and spotting gestures.

While Maestro's design was guided by the aforementioned ideals, the specifics of the design evolved according to an iterative design process. This process began with numerous mock-ups evaluated by open-ended interviews with potential users. This was followed by several Wizard of Oz simulations [1]. Working prototypes were then developed, and were tested in the laboratory by six individuals. The design was modified on numerous occasions in response to observations and feedback obtained in these laboratory tests. The remaining sections of this chapter describe Maestro's final design in detail, and are organized according to the three design goals as outlined above.

4.2 Software features and the gesture language

The structure of Maestro presentations is very similar to those of other contemporary presentation systems (PowerPoint, Keynote, Impress, etc.). Each presentation is composed of a sequential deck of slides, where each slide can contain some combination of written text, bullet hierarchies, and embedded figures. In support of the first design goal, Maestro allows presenters to use hand gestures to navigate the slide deck and to interact directly with the content of each slide. Maestro's navigation gestures provide both sequential and random access to slides. Maestro's content gestures support the communicative needs of the presenter (e.g., highlighting talking points), and allow presenters to adapt their presentations in response to audience questions and feedback (e.g., revealing additional details in response to an audience question). Both classes of gestures are designed to reflect the findings of the observational study, and to be easily detected by software. In the sections that follow, we first present an overview of the gesture language, and then provide more details regarding Maestro's navigation and content gestures.

4.2.1 Maestro's gesture language

In accordance with the recommendations outlined by the observational study, Maestro's gestures are designed to be performed by a presenter standing just outside the left edge of the projection screen. Gestures can be performed with either hand, and do not depend upon hand posture. In all cases, Maestro's gestures are designed to be performed quickly and without requiring much precision – a property recommended by Baudel in [2]. Consequently, gestures tend to consist of brief emphatic motion resulting in highly linear hand trajectories.

While Maestro's gestures are simple, and are quick to perform, they must be designed to prevent the accidental recognition of spurious commands. Consequently, Maestro's gestures are engineered to give rise to various non-accidental motion features. Non-accidental motion features, like non-accidental image features, are often

[1] In a wizard of Oz simulation, complex interactions are simulated by having an unseen individual control the software in response to actions performed by the user.

invoked in the field of computational perception [18, 31]. In general, non-accidental features are those which are best explained by underlying regularities or structures in the world rather than by coincidences. For example, coterminating edges in an image are often indicative of corners. Similarly, various properties of hand motion are often indicative of an intentional gesture [12]. Maestro's gestures make use of the following non-accidental motion features:

- Axis-aligned, linear hand motion
  In other research, we have shown that individuals are quite accurate when instructed to move their hands either horizontally or vertically, but that this axis-aligned linear motion does not typically occur in more natural unconstrained circumstances [12]. Consequently, Maestro's gestures consist of either horizontal or vertical strokes.

- Bimanual Interaction
  Simply stated, it is highly unlikely that two hands will accidentally move together along parallel lines, move apart collinearly, or rendezvous in space and time. Maestro uses this fact to its advantage when spotting gestures.

- Spatial Context
  To the extent possible, Maestro uses the content of the projected slides to contextualize the hand's motion. For example, observing a hand stop or change directions upon reaching a bullet point is a good indication that a gesture is imminent or occurring.

- Dwelling
  Dwelling, or holding a stationary position for a period of time, is one of the most commonly used cues for identifying the start or end of a gesture in other systems. However, dwelling is used sparingly in Maestro, and mainly in those instances where the presenter is unlikely to be addressing the audience.

While these features have been discussed individually, they are often combined to provide stronger cues for spotting gestures. For example, many of Maestro's gestures involve the rendezvous of two hands directly over a bullet point or figure, followed by the vertical motion of one or both hands. In combination, these cues provide strong evidence of a gesture occurrence.

The strategy of using non-accidental features to help spot gestures is generic, and applies to all of Maestro's gestures. This includes all gestures that support navigation and as well as those that support interactions with slide content. We now describe each of these gesture classes in turn.

4.2.2 Navigation gestures

Maestro's navigation gestures allow presenters to move between slides, to randomly access slides, and to scroll slides. Invariably, these gestures are performed in Maestro's staging area; a region which occupies the left margin of each slide. The staging area provides a spatial context for navigation gestures, clearly indicating that the object being manipulated is the slideshow itself. Early versions of Maestro did not use an explicit staging area, and presenters often found themselves performing these gestures in the slide's empty margins in order to avoid their confusion with gestures that operate on slide content. Maestro's staging area is simply a formalization of this practice, enhancing the slide's left margin to contain visual guides to help the presenter form gestures. In order to minimize its impact on the visual appearance of the slide, the staging area appears only when the presenter stops moving their hand within the slide's margin; however, gestures can be performed even when the stage is not visible.

The staging area is used almost exclusively for slide navigation. The simplest navigation gestures are "next slide" and "previous slide". To move to the next slide, a presenter places one hand in the center of the staging area and moves the hand straight down (figure 4.1a). Likewise, to move to the previous slide, a presenter need only move their hand straight up, again starting from the center of the staging area (figure 4.1b). A set of horizontal ruled lines in the staging area delineate the regions for invoking these gestures.

In addition to the "next slide" and "previous slide" gestures, Maestro also allows presenters to open a carousel containing thumbnails of all slides in the presentation. To access the carousel, the presenter places both hands in the stage's center section, and then pushes the hands away from their body (figure 4.1f). The carousel occupies the space vacated by the slide. Using deictic gestures, the presenter is then able to randomly select any slide in the carousel.

Finally, unique to Maestro is the ability to navigate *within* slides; Maestro allows presenters to author scrollable slides whose content is longer than the height of the projection screen. Scrollable slides address a well-known issue which arises because electronic slides provide only a limited space in which presentation authors may layout their material. In fact, the average slide contains only 40 words [58]. Edward Tufte, an expert in information visualization and one of PowerPoint's chief critics, has written about this issue:

> Many true statements are too long to fit on a PowerPoint slide, but that does not mean we should abbreviate the truth to make the words fit. It means we should find a better way to make presentations. [58, p. 1]

By allowing slides to scroll, the divisions between Maestro's slides is dictated by the nature of the material being presented rather than by screen real estate constraints. To scroll a slide downward, presenters begin by placing both hands in the stage's center region. They then move one of the hands straight down (figure 4.1d). This gesture is nearly identical to the "next slide" gesture, but is differentiated by the use of *two* hands. The slide responds by immediately scrolling down, and continues to scroll down as long as the hands remain in that particular configuration. The scroll speed is determined by the distance between the hands. Scrolling up is performed with a similar gesture (figure 4.1c).

Earlier versions of Maestro employed a manipulative slide scrolling gesture where the slide's vertical motion directly matched the presenter's hand motion. This panning gesture gave the illusion that the presenter was moving the slide up or down on the surface of the projection screen. Unfortunately, users complained that scrolling long distances required numerous strokes, which "felt like swimming" and was tiring. This is an example of a common touch-screen gesture that is ill-suited for large displays.

4.2.3 Slide content gestures

Maestro also affords direct interaction with the actual content of the slides. These gestures support the *communicative* needs of the presenter, and allow the presenter to adapt slide content in response to specific needs and circumstances that arise while presenting. First, blocks of text are automatically and instantly highlighted whenever one hand passes within their proximity (figure 4.2a). This serves to connect the verbal discussion with specific elements of the visual presentation. By pointing, and dwelling, presenters can follow hyperlinks incorporated in text, enabling them to access additional slides that explore a particular subject in more detail. Presenters can also selectively enlarge figures embedded alongside text. When enlarged, a figure occupies the entire screen, making even small details visible to the audience. To zoom into a figure, the presenter moves both hands into the figure, then pulls them apart vertically (figure 4.2e). These capabilities – highlighting points and enlarging figures to introduce more detail – were directly inspired by practices identified in the observational study.

Finally, presenters can author slides with hierarchical lists of bullets, with child bullets initially hidden. This capability allows the presenter to cater the detail of the presentation to the particular needs of the audience. For example, a bullet hierarchy may be expanded in order to reveal more details of the topic in response to an audience question. Alternatively, bullet hierarchies may start collapsed in support of short talks (e.g., when presenting a summary in a conference), only being expanded when presenting a longer version of the material (e.g., when giving a lecture). To reveal child bullets, the presenter places both hands next to the bullet point of interest, and slides one hand down. The reverse motion hides the child bullet point. These bullet gestures are similar to the scrolling gestures, differing only by the spatial context in which the gestures are performed.

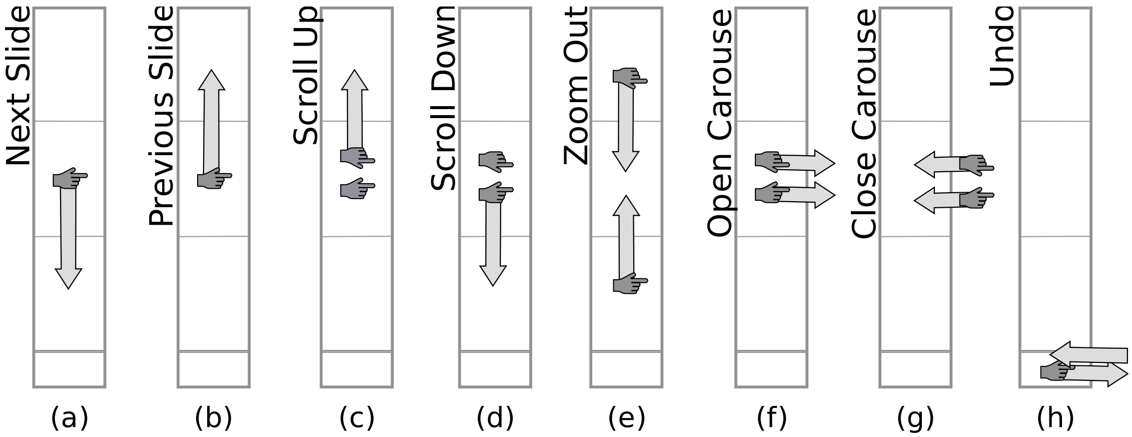


Figure 4.1: Maestro's presentation navigation gestures. Each of these gestures is performed in the staging area.

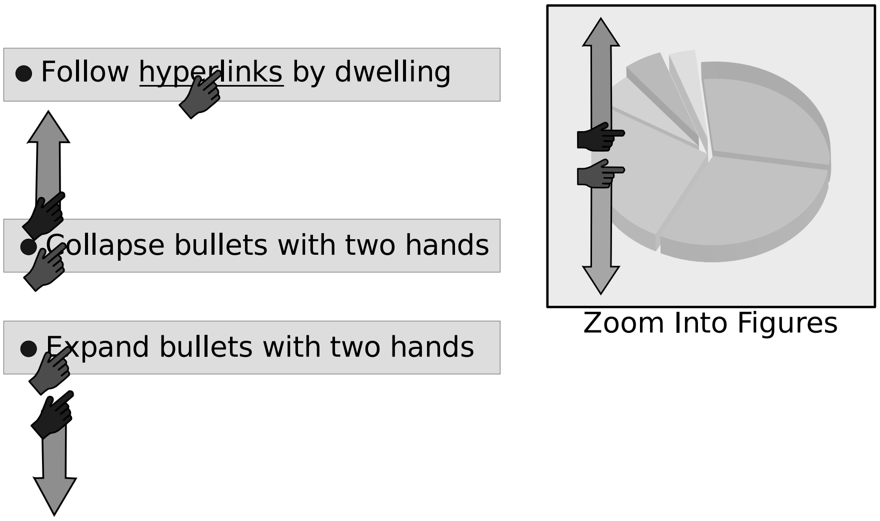

● Follow hyperlinks by dwelling
● Collapse bullets with two hands
● Expand bullets with two hands
Zoom Into Figures

Figure 4.2: Maestro's content gestures. Each of these gestures is performed in close proximity the object being operated upon.

4.3 Command affordances, feedback, and error recovery

One of the major challenges in developing Maestro was designing a mechanism for communicating command affordances and system feedback to the presenter. In the next sections, we describe how Maestro addresses each of these challenges in turn.

4.3.1 Command Affordances

Maestro communicates command affordances via a pair of cursors which follow the hands as they move around onscreen. At the most basic level, the cursors reveal where the system thinks the presenter is pointing. The cursors are augmented with *gesture mnemonics*, which serve both to indicate *which* commands are available in a particular context (similar to context-sensitive mouse cursors), and to remind users *how* to perform their gestures (figure 4.3). It is this latter purpose which is perhaps more interesting, and more specific to our particular application. As an example, when the hand is placed in the staging area of a slide, the cursor will be decorated with a mnemonic reminding the presenter that there is a previous slide that can be accessed by moving the hand upward. This same mnemonic does not appear when visiting the first slide of the presentation since it is not possible to go to a "previous" slide in this situation. Mnemonics are not meant as detailed gesture instructions, but instead serve to indicate the general direction and form of the gesture.

In addition to gesture mnemonics, Maestro uses various other simple visual cues to indicate command affordances. For example, the tops and bottoms of slides have rounded corners in order to indicate if they can be scrolled. If a slide can be scrolled down, the bottom rounded corners are not visible. This indicates that the bottom of the slide has not yet been reached. Similarly, the top rounded corners are not visible when the slide can be scrolled upwards.

4.3.2 Command Feedback

Since Maestro relies entirely on computer vision for input, tactile and other forms of feedback are not available. Consequently, Maestro renders all feedback to the display. Since the display is shared between the audience and the presenter, and because feedback is directed only at the presenter, all visual feedback must be kept quite subtle. For feedback, Maestro displays translucent icons within the staging area to reassure the presenter that a command has been received. These icons remain displayed for several seconds allowing the presenter ample time to find them. Because the icons fade out with time, presenters can assess their relevancy.

One important lesson learned about providing feedback in a gesture-based setting is that the system should provide feedback even when gestures are performed

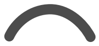
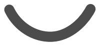
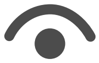
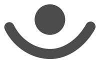
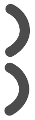
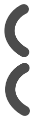
Gesture	Mnemonic
Previous Slide	
Next Slide	
Scroll Up	
Scroll Down	
Open Carousel	
Close Carousel	

Figure 4.3: Several gesture mnemonics used by Maestro. The dots in the "scroll up" and "scroll down" mnemonics indicate the presence of a stationary hand.

in invalid contexts. It is often tempting to use context and system state to rule out as many gestures as possible. For example, one could argue that the "scroll down" gesture need not be considered when the system is displaying a figure in the full screen (since, by definition, the figure is fitted to the screen and need not be scrolled). While this strategy improves efficiency and helps to reduce false-positive rates, it precludes the possibility of providing negative feedback to the user. Lack of feedback is almost always attributed to a recognizer error (specifically a false-negative), and the user will often repeat the gesture in error. A more appropriate response is to have the system acknowledge the gesture but provide some indication of the problem.

4.3.3 Error recovery

As suggested by both Baudel [2] and Cao [5], Maestro allows presenters to recover from recognition errors by issuing an "undo" command. Maestro's undo command can be accessed at any time by moving the hand as depicted in figure 4.1h. Importantly, the staging area always displays the name of the last command recognized, so that the presenter will know which command will be undone by the aforementioned gesture.

In addition to supporting the undo operation, Maestro allows slides to be navigated using common keyboard commands (e.g., left-arrow, right-arrow). Maestro also interfaces with many presentation remote controls. This provides presenters with a “fail safe” allowing them to continue a presentation in case of technical difficulties with Maestro’s computer vision and gesture recognition machinery. In support of the “fail safe” mode, context-sensitive commands can be issued using the mouse.

4.4 Hand tracking and gesture spotting

In the sections that follow, we describe Maestro’s gesture recognizer, along with the features that facilitate differentiating gestures from gesticulation. However, before gestures can be spotted, the user’s hands must be detected and tracked. We describe the hand tracker below, followed by a description of the gesture recognizer.

4.4.1 Hand tracking

The first step in recognizing gestures is to detect and track each of the presenter’s hands. It turns out that this task is rather challenging in a front-projected environment since the light emanating from the projector can vastly alter the appearance of the hands. Consequently, detecting hands using only skin color or shape cues is rather challenging [14, 55, 30]. Motion detection, using background subtraction techniques, is also challenging since the presentation is dynamic and the background is always changing. Of course, the background is controlled by software. This leaves open the possibility of performing known-background subtraction in order to detect occluding objects [14, 30]. However, this too is non-trivial since it requires calibrating the projector for color constancy, and requires modeling the camera’s color response. This is both complex and computationally expensive.

In the interest of reliably detecting the presenter’s hands in real-time, *and because hand tracking is not the primary focus of our research*, Maestro facilitates hand tracking by requiring users to wear a mismatched pair of brightly colored gloves. Specifically, one glove is bright red (or orange) and the other glove is light blue (or cyan, which reflects blue and green light about equally). The use of bright, high-saturation colors is important because dark colors can easily be confused with the shadows cast by the hands or arms, while low-saturation colors tend to be easily distorted by the light emanating from the projector. When presenters wear the gloves, hand detection can proceed using simple color thresholding techniques that are computationally inexpensive. To further simplify hand detection, the presentation system renders all slides in grayscale. This ensures that the gloved hands are not confused with elements of the projected slides.

Even in this idealized hand-tracking environment, hand tracking is non-trivial. One notable challenge is that the hands tend to “disappear” when they pass over especially dark regions of the slides; since projectors render dark regions by limiting the amount of light that reaches the screen, the hands are often poorly illuminated when in these dark regions. This problem typically occurs when gestures operate on dark images. To overcome this difficulty, Maestro temporarily lightens the appearance of dark images when it senses the hands approaching.

While Maestro's hand tracking system is certainly not ideal, it is fast, it is reliable, and it enabled us to develop Maestro to the point where it could be used on a day-to-day basis. Maestro's hand tracker can be easily replaced by a more sophisticated tracker should one become available; this should not affect the functioning of the gesture recognizer which is described in the next section.

4.4.2 Gesture spotting

Another challenge faced when designing Maestro was establishing gesture recognition machinery capable of spotting meaningful gestures that are embedded in longer sequences of hand motion. Gesture spotting is especially challenging in a presentation environment where commands are issued only intermittently, meanwhile the presenter may make use of a great deal of gesticulation while discussing the slide content. Here, and elsewhere in Part I of this document, we describe how gesture spotting was achieved using Maestro's "ad-hoc" gesture recognizer. This recognizer was developed in order to support the rapid prototyping of numerous gesture languages, and is supplanted in Part II of the thesis by a more sophisticated recognizer. However, a brief description of the ad-hoc recognizer is important because it was used both throughout Maestro's design process, and in Maestro's real-world evaluation.

With the ad-hoc recognizer, gesture spotting occurs in two steps. First, the recognizer identifies an instantaneous cue demarcating either the start or the end of a gesture (or sometimes both). Examples include observing two hands rendezvous over a specific bullet, figure or region. This is followed by the recognition of the gesture's motion in space. In this sense, the starting or ending cues (also known as segmentation cues) serve to initiate a local search for the gesture. In both stages of the recognition process, various properties of the hand trajectories (e.g., start/end location, path length, general direction of travel, moment of inertia) are measured and are tested against gesture templates containing manually established range-constraints for these properties.

4.5 Discussion

In this chapter, Maestro's design was described in detail, and represents one possible solution to the design challenges. In describing Maestro's design, we have been careful to list our motivations and the lessons we learned. We hope these are useful to others exploring similar research. Nevertheless, it is our belief that Maestro is at least representative of gesture-based presentation systems in general, and that it can be used to evaluate the viability of gesture-based presentations in a real-world day-to-day context; Maestro supports a superset of the features enabled by similar systems in the literature, and the entire system was as carefully and painstakingly designed using an iterative design process. In the next chapter, we describe the results of Maestro's real-world evaluation.

Chapter 5

Evaluation and Lessons Learned

To assess the viability of a gesture-based presentation system, we deployed Maestro in a classroom for several weeks. During this time, Maestro replaced PowerPoint as the main presentation system. The data collected from this trial consisted of 12 hours worth of lectures, and represents the most extensive real-world evaluation of a gesture-based presentation system available in the literature. In this chapter, we discuss the results of this experiment.

5.1 Study Overview

Our deployment study was designed to assess the overall viability of a gesture-based presentation system in a real-world setting. In particular, we sought to answer the following questions:

- How does gesture-based input compare to more traditional input modalities such as keyboards, mice and presentation remotes?
- What software features are most useful, and which need further refinement?
- How does gesture-based input fit in with current presentation practices? Does gesture-based input noticeably alter the dynamics of presentations?

To answer these questions, a research supervisor used the system to give lectures to approximately 100 students over a two-week period. The lectures were part of a third-year university course unrelated to the research project. During this period, Maestro was used a total of 12 times to deliver six unique one-hour lectures (lectures were given three times a week, with the same lecture given twice a day). For each pair of lectures, the lecturer carried in, set up, and calibrated the necessary equipment for deploying Maestro. In this case, the equipment included a laptop, an external web camera, and the colored gloves; the room was already equipped with a non-portable data projector. Since the classroom was used by other courses, Maestro's portability and ease of deployment was a necessary precursor to these trials.

As mentioned above, lectures were taught by one of Maestro's researchers. This researcher functioned in a supervisory role during Maestro's development, and he was not familiar with its specific implementation. Accordingly, he had to learn how to setup, calibrate, and use the system, as well as author content. Thus, while he was involved in the project, his experiences in using the system were closer to those of a first-time user; in fact, there were many times when he needed to ask what features were available and how they were used.

Prior to deploying Maestro, lectures were given for approximately eight weeks using PowerPoint controlled by a laptop keyboard. The laptop was located at a lectern in a corner of the classroom. The blackboard was also used occasionally during this time. After Maestro's deployment, lectures were given for two weeks using PowerPoint and a wireless remote control. While this evaluation did not attempt to perfectly balance the use of the various interaction mechanisms, it nonetheless serves to provide the first real-world comparison of three distinct control mechanisms, and includes the first longitudinal evaluation of a gesture-based interface to a presentation system.

For data collection, several of the lectures were videotaped by the author of this document, who also took notes. Students were encouraged to provide feedback during lectures and were given a questionnaire at the end of the term to provide both structured and open-ended feedback. We begin by describing the survey results, and then discuss observations compiled from the videos, written notes, and audience feedback.

5.2 Survey Results

After completing several weeks of lecturing using each of the four presentation styles (blackboard, PowerPoint controlled with the keyboard, PowerPoint controlled with a wireless remote, and Maestro), students were asked to complete a questionnaire consisting of 40 Likert items. A Likert item is a written statement, paired with a discrete symmetric bipolar set of responses with which the participant self-reports their agreement with the statement [27]. Maestro's audience questionnaire used a 4-point response set ranging from 1 (strongly disagree) to 4 (strongly agree). While a five-point set is more common (which includes a "neutral" option), we removed the neutral option so as to force participants to indicate either a positive or negative expression of agreement to each statement.

Importantly, the questionnaire was divided into four themes: a comparison of the various presentation styles; an evaluation of Maestro's features; an evaluation of Maestro's visual appearance; and finally, an evaluation of Maestro's gesture recognition machinery. Responses in each of these categories are detailed in the sections that follow. Before reporting the results of the survey, it is worthwhile reviewing the statistical methods that were utilized in the analysis.

5.2.1 Statistical Methods

Audience members completed the questionnaire voluntarily. Approximately 70 of more than 100 students completed the questionnaire. After administering the survey, the responses were compiled so as to gauge audience agreement with each statement, and to compare their responses across pairs of statements. In this setting, selecting an appropriate statistical test is a matter of some debate. Since Likert scales are presented as symmetric and bipolar continuums, it is common to treat the responses as interval data [27]. Provided that the distribution of the responses is approximately normal (or at least, bell-shaped), a t-test can be used to gauge agreement with any single statement, and a paired t-test can be used to compare responses to a pair of statements.

However, the responses to the survey's Likert items are certainly not actually normally distributed since they are discrete and are selected from a small set of possible values [27]. In this sense, the responses are more correctly labeled as ordinal data as opposed to interval data. Moreover, histograms of audience responses are often not bell-shaped (as in figure 5.2b) suggesting that a t-test would be inappropriate in many cases. Finally, treating the responses as interval data (as opposed to ordinal) assumes that the responses are somehow equidistant from one another. However, this assumption is not always justifiable. In our case, it suggests that the distinction between "strongly agree" and "agree" is similar to the distinction between "agree" and "disagree"; but, it is unlikely that these differences are directly comparable in this way.

As a consequence of the aforementioned properties of the data, it is technically more appropriate to use a nonparametric statistical test in our analysis [27]. In order to accomplish this, audience responses are collapsed into a nominal form where each participant either expresses positive or negative agreement to each statement. In other words, participants either "accept" or "reject" each statement. In this way, each audience response can be modeled as a single Bernoulli trial in which a "success" corresponds to acceptance of the statement. The total number of successes to any statement follows a Binomial distribution. A *sign test* can then be used to both gauge overall agreement with a single statement, and to compare responses to a pair of statements. When gauging agreement to a single statement, the null hypothesis proposes that the responses are from a Binomial distribution in which the probability of success is $p = \frac{1}{2}$. Accepting the null hypothesis suggests a strong possibility that audience opinion is evenly divided on the statement. Similarly, when comparing responses to a pair of statements, responses to one statement are subtracted from responses to the other statement on a per-participant basis (i.e., responses are paired). In cases where this difference is zero (i.e., no preference is stated), the response-pair is simply ignored. Cases where the difference is *positive* are considered to be "successes". Again, the number of successes follows a Binomial distribution, and the null hypothesis assumes that the probability of success is $p = \frac{1}{2}$.

In this chapter, the sign test is used as the main statistical method for data analysis. However, the results of t-tests are also reported as a further descriptive measure of audience agreement. *The t-test results should be considered with some skepticism.* While the t-test may not be technically appropriate, it nonetheless provides useful data. Specifically, the t-test utilizes the magnitude of the audience responses (i.e., "agree" vs. "strongly agree"), while the sign test does not. In practice, we have found that both tests reveal similar trends in the data, but that the sign test is far more conservative when computing statistical significance.

5.2.2 Comparing presentation media

The first section in the questionnaire sought to compare Maestro with use of a blackboard; PowerPoint with a keyboard and mouse; and, PowerPoint controlled by a wireless remote. Participants rated each system independently in terms of interactivity, visual appeal, and efficiency. This portion of the questionnaire was very similar to the one used by Cao *et al.* in [5].

To analyze the data from this section of the questionnaire, we used a paired sign test. Recall, this statistical test directly compares an individual's perceptions of one presentation medium to their perceptions of another presentation medium. In comparing the competing presentation technologies to Maestro, we found the following results (at a significance level of $\alpha = 0.05$):

- Maestro is considered more interactive than using the blackboard ($p < 0.001$), PowerPoint with a keyboard ($p < 0.001$), and PowerPoint with a remote ($p < 0.001$).
- Maestro is considered more visually appealing than using the blackboard ($p < 0.001$). No statistical difference was found when comparing the visual appeal of Maestro to that of PowerPoint using a keyboard ($p = 0.664$) or PowerPoint with a remote ($p = 0.832$).
- Maestro is seen as less efficient than PowerPoint using a remote ($p < 0.001$). No statistical difference was found between Maestro and the blackboard ($p = 0.627$); or between Maestro and PowerPoint controlled using a keyboard ($p = 0.076$). However, a low p-value in the latter case suggest a trend towards finding Maestro less efficient than a keyboard-controlled PowerPoint presentation.

The mean scores across these dimensions and presentation media are presented in figure 5.1.

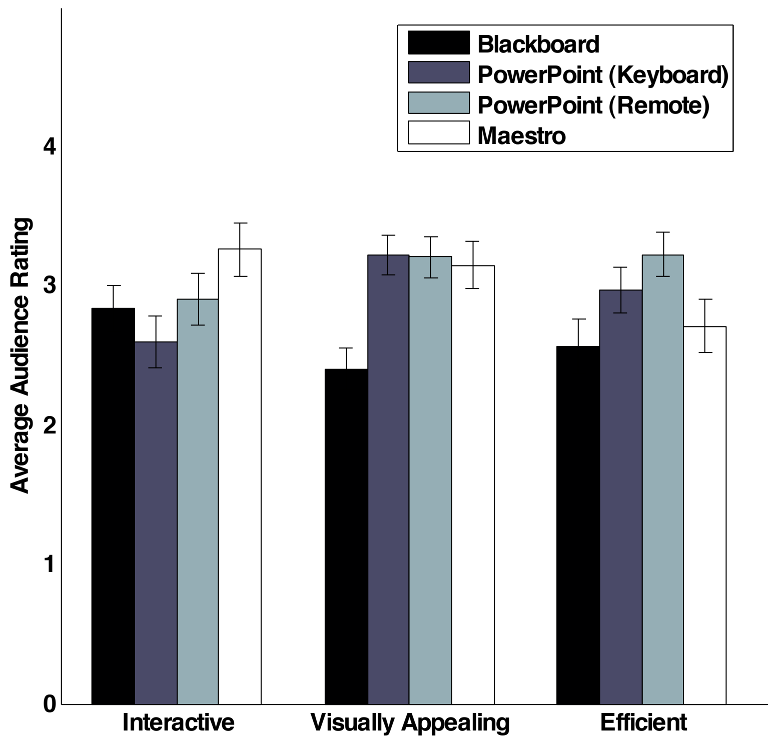


Figure 5.1: Mean scores for each of the presentation media. Error bars represent a 95% confidence interval about the sample means.

From these results we find that Maestro is considered to be more interactive than the other presentation media and input modalities. This result validates the notion that gesture-based input can positively enhance presentations. At the same time, Maestro was found to be less efficient than PowerPoint. This lower efficiency score is worthy of further investigation, but there are a number of potential reasons for this lower score. First, advancing slides requires a relatively large physical action; the presenter must orient himself next to the projected content, position the hand, then swipe it downward. This takes quite a bit longer than pushing a button on a remote that is already in hand. Also, the lower perceived efficiency could be partially attributed to delays caused by occasional gesture recognition errors. More work is required to determine the importance of these contributing factors.

5.2.3 Evaluating Maestro's features

The second section of the questionnaire asked participants to rate Maestro's specific software features. Since audience participation was voluntary, the number of responses varied from 62 to 64 on any given item. Detailed results of these responses are presented in figures 5.2a - 5.2c, and in table 5.1b.

In regards to Maestro's specific software features, Maestro's ability to present figures in full-screen mode was overwhelmingly welcomed by participants: 42% of the students *agreed*, and 52% *strongly agreed*, with the statement that "it is often useful to view figures in full-screen mode." The positive response to this feature is highly statistically significant, and represents the most positive response to any statement in the survey.

The majority of participants also responded positively to the automatic highlighting of bullet points, with 64% indicating it was a useful feature. Again this result is statistically significant at the 5% level (p-value of 0.033). Although the feature elicited a positive response, there are many opportunities for improvement. For example, one student commented that the bullet highlighting decreased readability because it placed a gray background behind black text, thereby reducing contrast. This is a legitimate concern that could be addressed by reversing foreground and background colors when highlighting bullet points, or by using some alternative approach to emphasizing text. Additionally, one student noted that the system should also allow the presenter to highlight individual keywords and phrases *within* bullet points. This is certainly an interesting possibility, and is consistent with the behaviors witnessed in the observational study. More research must be done to determine how best to integrate this feature.

Audience members were also asked to evaluate the usefulness of scrollable slides. Here, only about 42% of participants thought that the ability to scroll slides up and down was advantageous. However, with a p-value of over 0.250, the null hypothesis cannot be rejected. In other words, there is a strong possibility that the audience opinion of this feature is evenly divided. In either case, the superiority of scrollable slides was not accepted by the audience. In terms of open-ended feedback, all comments regarding this feature were negative. For example, one student wrote:

> I really don't like the scrolling slide feature. To me slides are meant to have concise information on a single point (...) Scrolling is sort of going against the strengths of the medium.

One important note about scrollable slides is that this feature was used only occasionally during the evaluation period. In part, this is because most of the slides were transcribed from existing PowerPoint presentations. Since PowerPoint doesn't support this feature, slide content almost always fit on a single screen. A few slides did require scrolling, but only because font and layout differences between PowerPoint and Maestro caused some bullets to not fit on a single page. Consequently, scrolling only ever revealed one or two extra lines of text, and was not used to its full potential. This fact was noted by one student, who commented that "scrollable slides might be better if there is more content."

Finally, expandable bullet points and hyperlinks were rarely used during the trial, and were not represented in the audience questionnaire. Again, these features were infrequently used because slides were transcribed from existing PowerPoint presentations (which do not support these features). Additionally, while the slide carousel was always available for use, this feature was never utilized during the two-week trial; there was little need to randomly access slides once the presentation was started.

S1. “The automatic highlighting of bullet points, whenever the hand is nearby, helps clarify which point the presenter is discussing.”

S2. “It is often useful to view figures in a full-screen mode.”

S3. “Slides that can be scrolled up or down are advantageous since they can contain more material than can be displayed at once.”

(a) Survey statements

	Descriptive Statistics		Sign Test		T-Test		
	Median	Mode	% Successes	p-value	$\hat{\mu}$	$\hat{\sigma}$	p-value
S1.	3 (agree)	3 (agree)	**64.1%**	**0.033**	2.70	0.89	0.071
S2.	4 (strongly agree)	4 (strongly agree)	**93.8%**	**< 0.001**	**3.44**	**0.66**	**< 0.001**
S3.	2 (disagree)	2 (disagree)	41.9%	0.253	2.29	0.86	0.059

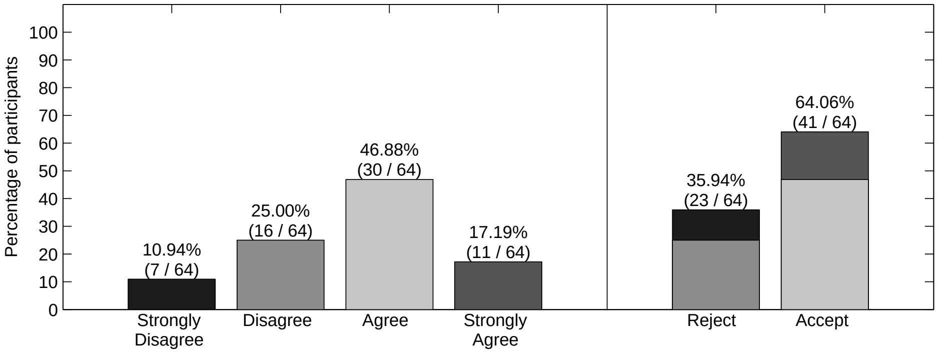
(b) Descriptive statistics and statistical test results of survey responses.

Table 5.1: Survey results for statements about Maestro’s features. Items in bold correspond to statistically significant results.

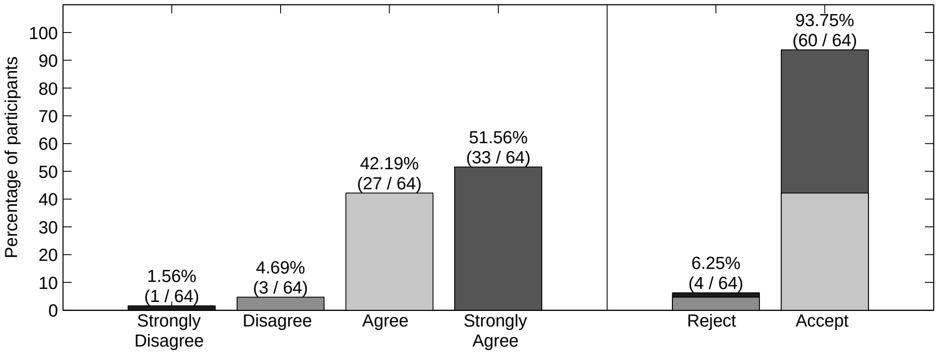
5.2.4 Evaluating visual appearance

The third section of the questionnaire assessed Maestro’s visual appearance. Results are depicted in figures 5.3a - 5.3f, and in table 5.2b. When developing Maestro, there was some concern that the staging area would detract from the presentation’s visual appeal. However, about 63% of the audience reported that the staging area was not a problem. While this result is not quite statistically significant when using the sign test, its low p-value suggests a positive trend (p-value = 0.060). This is supported by the t-test, which did find statistical significance (p = 0.025). Excluding the staging area, 85% of the audience found Maestro’s slides to be visually appealing. In terms of audience comments, one student wrote that we “could improve on the look, in terms of colors”, but “the layout, in general, is well designed.” These findings are not surprising given that Maestro’s slides are very similar to those of PowerPoint and similar systems.

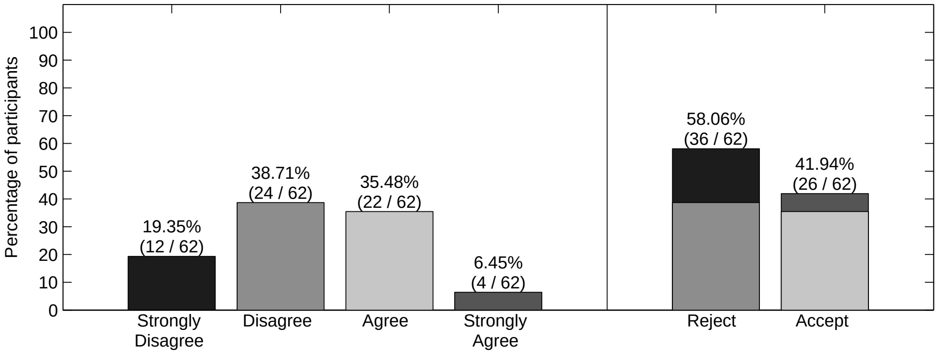
The third section of the questionnaire also sought to gauge audience opinion on the use of colored gloves. While most of the audience found both the hand tracking cursors and the colored gloves distracting, the results are not quite statistically significant (61%, $p = 0.098$ for the cursors; and 62%, $p = 0.060$, for the gloves). Again, a relatively low p-value, and the t-test results, suggest a trend towards



(a) S1: “The automatic highlighting of bullet points, whenever the hand is nearby, helps clarify which point the presenter is discussing.”



(b) S2: “It is often useful to view figures in a full-screen mode.”



(c) S3: “Slides that can be scrolled up or down are advantageous since they can contain more material than can be displayed at once.”

Figure 5.2: Responses to various statements regarding Maestro’s features.

finding the colored gloves distracting. In regards to the gloves and cursors, most audience comments were negative. For example, one student wrote “what I disliked most about Maestro was the glove coloring”, while another wrote: “I found the dots, which follow (the presenter’s) hand around, very distracting”.

Additionally, 60% of the audience found Maestro’s insistence on monochromatic slides to be less than ideal, but these results are not statistically significant ($p = 0.169$). Recall that Monochromatic slides are necessary in order to prevent the hand tracker from mistaking slide content for the colored gloves. As was noted by one of the students, the acceptability of monochromatic slides depends upon the material being presented.

Finally, this section of the questionnaire probed the audience’s opinion regarding Maestro’s bimanual gestures. In this case the responses were evenly divided: 32 participants accepted that bimanual gestures were “as natural as those that involve only one hand”, while another 32 participants rejected the statement. As for open-ended feedback, one student complained that “the use of 2 hands seems cumbersome”. This result is interesting because Meastro’s two-handed gestures, such as “zoom into figure”, were designed to closely match numerous naturally occurring gestures witnessed in the observational study (see Chapter 3). Even when based on natural gestures, half the audience found bimanual gestures to appear artificial.

These results clearly indicate the importance of moving to a hand-tracking technology that does not require the use of gloves or monochromatic slides. They also reinforce the challenge of providing affordances to the presenter without distracting the audience. In Maestro’s case, the context-sensitive cursors were not sufficiently subtle to achieve this objective.

5.2.5 Evaluating the gesture recognizer

The final section of the questionnaire sought to evaluate the perceived accuracy of Maestro’s gesture recognizer. Results are presented in figures 5.4a - 5.4e, and in table 5.3b. In this section of the questionnaire, we distinguish the “next slide” and “previous slide” gestures from the other gestures, because these two gestures correspond to the two most commonly used commands. The recognizer’s accuracy for the “next slide” and “previous slide” gestures was perceived by the audience as being “good” (68% rated it “good” or better, $p = 0.007$). However, about the same number of participants rated *the other* gestures as being recognized with “poor” or worse accuracy.

In assessing these results, we note that a controlled test of Maestro’s gesture recognizer found that 86% of gestures were correctly recognized for new users, increasing to 96% for expert users. In both cases, fewer than 1% of all gesture occurrences were false positives. These recognition results compare favorably to other gesture-based systems [2, 28], and are discussed in detail in Chapter 8. While we do not have recognition rates for the classroom deployment, the questionnaire

S4. “The *staging area* (a region to the left of every slide, where many gestures are performed) does not detract from the visual appearance of the slideshow.”

S5. “The layout of the slides, excluding the *staging area*, is visually appealing.”

S6. “The red and blue dots, which indicate where the presenter is pointing, are not distracting.”

S7. “The use of red and blue gloves is not distracting.”

S8. “It is acceptable that the slides are monochromatic (black and white) since color does not add much to the presentation.”

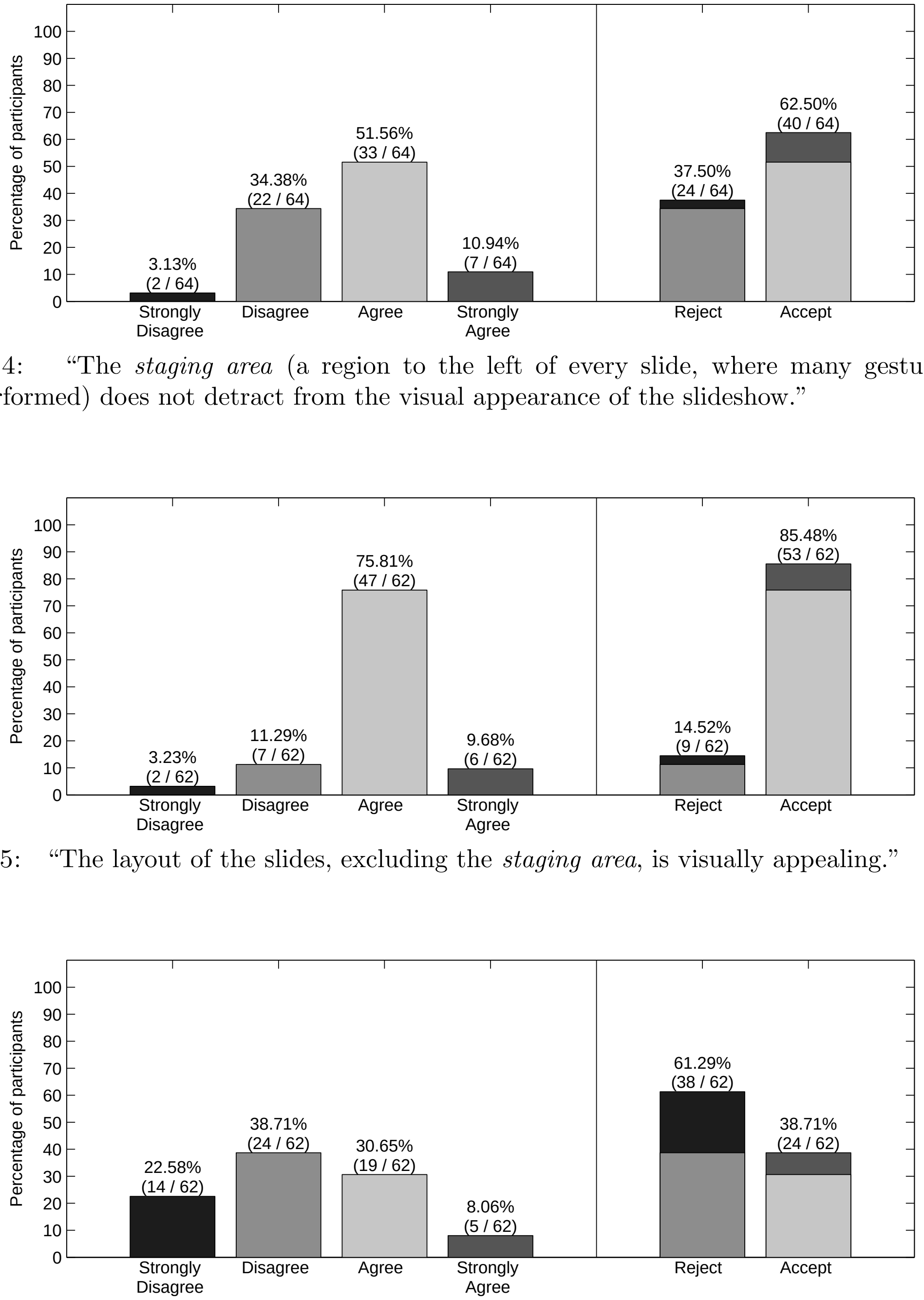
S9. “Maestro’s two-handed gestures appear just as natural as those that involve only one hand.”

(a) Survey statements

	Descriptive Statistics		Sign Test		T-Test		
	Median	Mode	% Successes	p-value	$\hat{\mu}$	$\hat{\sigma}$	p-value
S4.	3 (agree)	3 (agree)	62.5%	0.060	**2.70**	**0.71**	**0.025**
S5.	3 (agree)	3 (agree)	**85.5%**	**< 0.001**	**2.92**	**0.58**	**< 0.001**
S6.	2 (disagree)	2 (disagree)	38.7%	0.098	**2.24**	**0.90**	**0.027**
S7.	2 (disagree)	2 (disagree)	37.5%	0.060	**2.16**	**0.84**	**0.002**
S8.	2 (disagree)	2 (disagree)	40.6%	0.169	**2.28**	**0.72**	**0.018**
S9.	2.5 (divided)	2.5 (divided)	50.0%	1.00	2.48	0.64	0.846

(b) Descriptive statistics and statistical test results of survey responses.

Table 5.2: Survey results for statements about Maestro’s features. Items in bold correspond to statistically significant results.

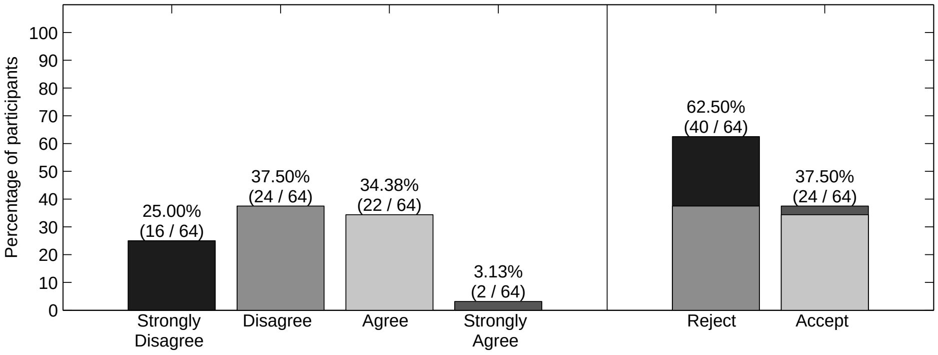


(a) S4: "The *staging area* (a region to the left of every slide, where many gestures are performed) does not detract from the visual appearance of the slideshow."

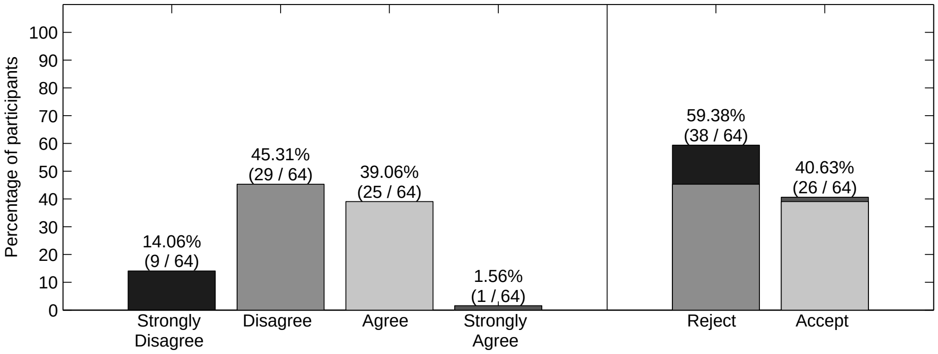
(b) S5: "The layout of the slides, excluding the *staging area*, is visually appealing."

(c) S6: "The red and blue dots, which indicate where the presenter is pointing, are not distracting."

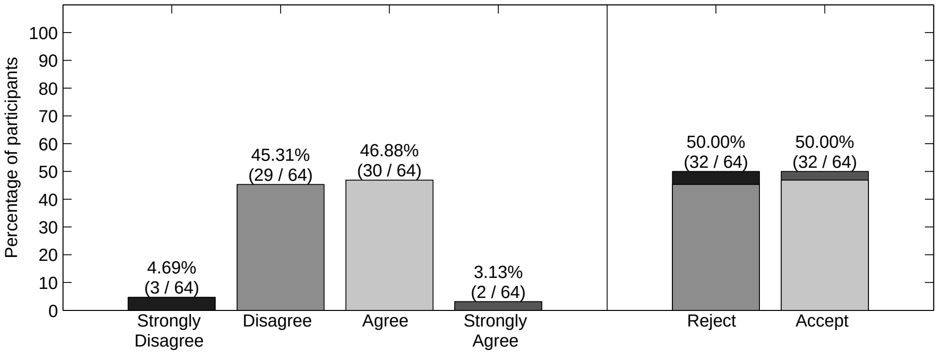
Figure 5.3: Responses to various statements regarding Maestro's appearance.



(d) S7: "The use of red and blue gloves is not distracting."



(e) S8: "It is acceptable that the slides are monochromatic (black and white) since color does not add much to the presentation."



(f) S9: "Maestro's two-handed gestures appear just as natural as those that involve only one hand."

Figure 5.3: (Continued) Responses to various statements regarding Maestro's appearance.

results indicate that recognition errors are quite distracting, and that a much better accuracy must be attained.

In terms of the types of errors that do occur, the audience perceived false negatives as occurring more frequently than false positives ($p = 0.047$). This echoes the quantitative recognition results reported above, and is an important result because we explicitly constructed the gesture recognizer to favor false negatives over false positives. The motivation for engineering this bias is well summarized by one student who commented:

> (False negatives) are almost never distracting because (it is) easy to resume (the) train of thought. (False positives) are frequently distracting because the class laughed and the presenter was oblivious (to the error).

We were also interested in the perceived accuracy of the hand tracking system. Participants were asked to rate the accuracy of the system when performing precision targeting tasks (such as pointing to text). Here 87% of the audience responded that the accuracy was good or better, representing a highly statistically significant result (p-value < 0.001).

Finally, participants were also asked to rate the responsiveness of the system. Specifically, audience members were asked to rate the latency between the performance of a gesture and the system response, and the latency between the hand motion and the cursor. In both cases, the system responsiveness was considered to be good or better, but results for the former suggest a trend rather than statistical significance (63%, $p = 0.056$; and 67%, $p = 0.011$, respectively).

S10. “The accuracy of the gesture recognizer when performing the *next slide*, and *previous slide* gestures.”

S11. “The accuracy of the gesture recognizer when performing other gestures such as *zoom into* and *zoom out of* figures, etc..”

S12. “The accuracy of the gesture recognizer when performing precision targeting, such as pointing to text.”

S13. “The delay between performing a gesture and the system’s response (e.g: changing to the next slide)”

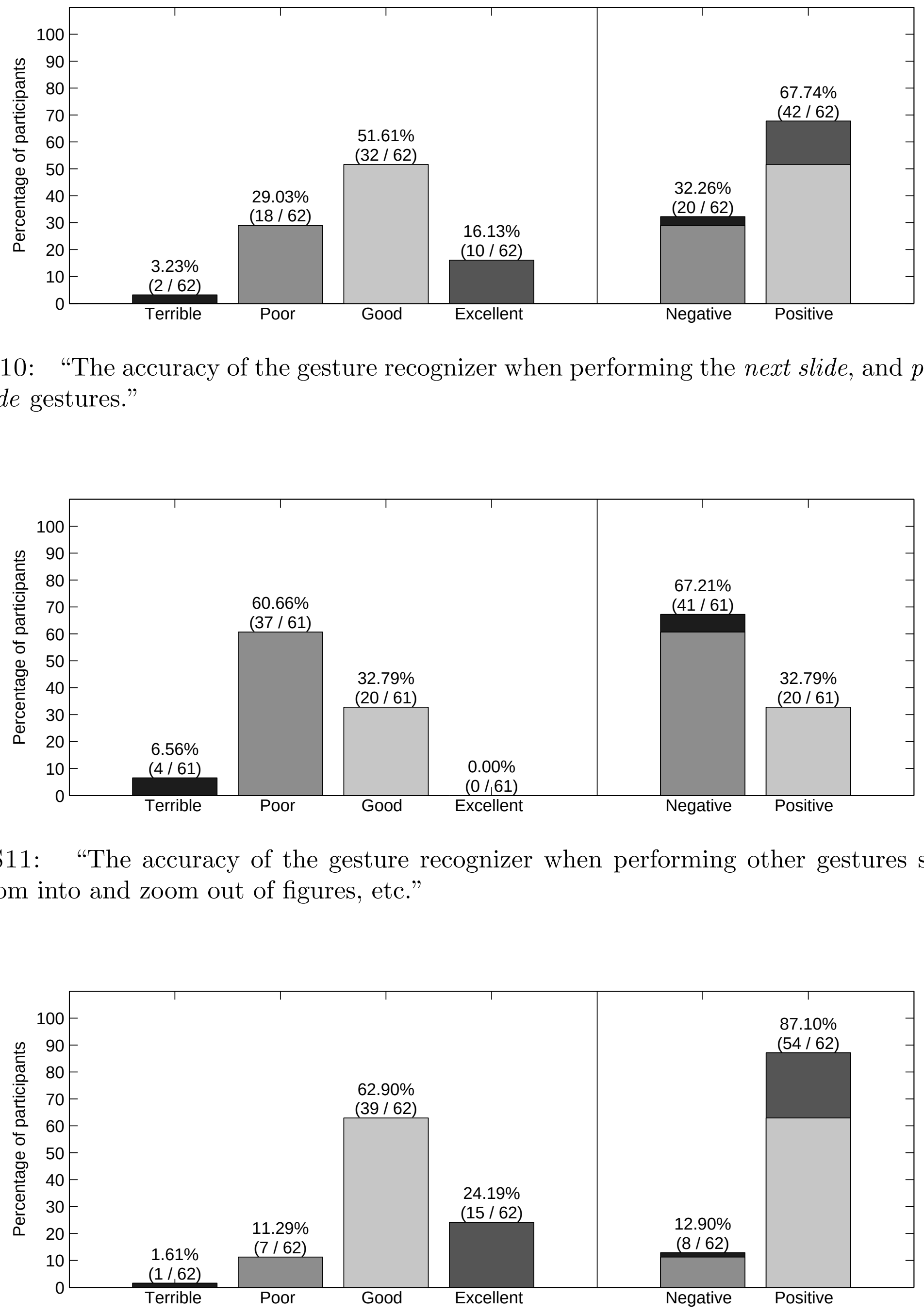
S14. “The delay between the hand motion, and the motion of the onscreen cursors.”

(a) Survey statements

	Descriptive Statistics		Sign Test		T-Test		
	Median	Mode	% Successes	p-value	$\hat{\mu}$	$\hat{\sigma}$	p-value
S10.	3 (good)	3 (good)	**67.7%**	**0.007**	**2.81**	**0.74**	**0.002**
S11.	2 (poor)	2 (poor)	**32.8%**	**0.010**	**2.26**	**0.57**	**0.002**
S12.	3 (good)	3 (good)	**87.1%**	**< 0.001**	**3.10**	**0.65**	**< 0.001**
S13.	3 (good)	3 (good)	62.9%	0.056	**2.77**	**0.77**	**0.007**
S14.	3 (good)	3 (good)	**66.7%**	**0.011**	**2.76**	**0.67**	**0.003**

(b) Descriptive statistics and statistical test results of survey responses.

Table 5.3: Survey results for statements about Maestro’s gesture recognizer. Items in bold correspond to statistically significant results.

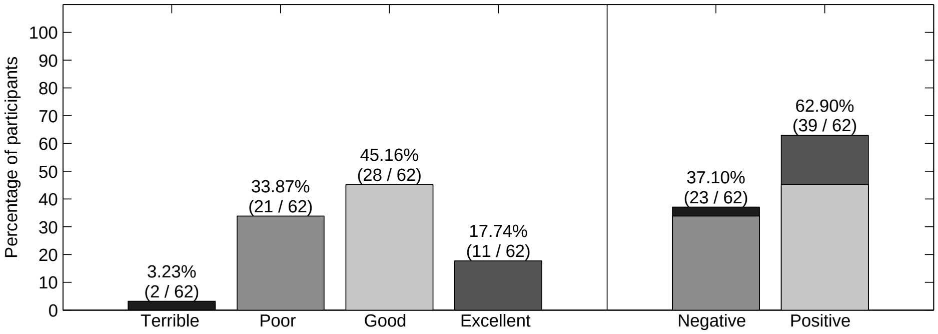


(a) S10: "The accuracy of the gesture recognizer when performing the *next slide*, and *previous slide* gestures."

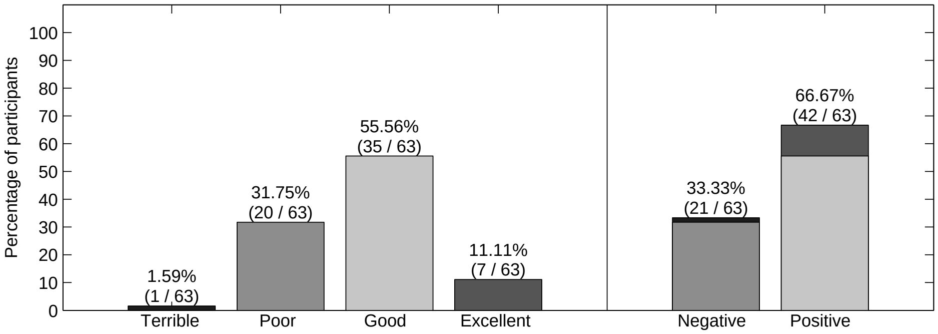
(b) S11: "The accuracy of the gesture recognizer when performing other gestures such as zoom into and zoom out of figures, etc."

(c) S12: "The accuracy of the gesture recognizer when performing precision targeting, such as pointing to text."

Figure 5.4: Responses to various statements regarding Maestro's gesture recognizer.



(d) S13: "The delay between performing a gesture and the system's response (e.g: changing to the next slide)"



(e) S14: "The delay between the hand motion, and the motion of the onscreen cursors."

Figure 5.4: (Continued) Responses to various statements regarding Maestro's gesture recognizer.

5.3 Observations and open-ended feedback

In addition to the statistical results derived from the audience questionnaire, we were also interested in the observations and feedback provided by the presenter, the audience, and the observer who was taking notes. We first present some of the more general comments provided by the audience. We then reflect on what the presenter found useful.

5.3.1 Audience feedback

Several audience comments, specific to certain features or properties of Maestro, have already been presented in the previous sections. This section lists more general audience comments regarding the use of Maestro as a presentation medium. In this regard, comments are both positive and negative, and closely match the conclusions derived from the survey results. Some of the more thoughtful comments are listed below:

> The system appeared to work fairly well, with some obvious issues with precision. The fact that the presenter needs to make large obvious movements and wear bright-colored gloves can be distracting from the actual content. I did like the ability to zoom in on images and highlight points.

> Is it really that much better than an average wireless remote? I haven't decided.

> (Maestro) allowed the presenter to present material without having to directly interact with the computer, and it has advantages over remote devices because of an increase in the range of functions.

> Having a remote to switch slides is sufficient (...) and just as effective as Maestro.

5.3.2 Useful software features

Echoing the questionnaire results and audience feedback, the presenter found the ability to selectively enlarge content and highlight talking points to be the two most useful features of Maestro. We describe how each feature was used in practice.

Figures were frequently embedded in slides next to bullet points. After giving an introduction to the slide's content, figures were often enlarged to full screen, enabling the audience to see greater detail as the presenter described specific elements of the figure. As used, this ability to provide an overview, then enlarge figures, afforded a "focus plus context" presentation style that was missed when moving back to PowerPoint slides. While similar effects could be achieved using PowerPoint (e.g., by scripting a series of animations), Maestro enabled this dynamic style of presentation to unfold at *presentation time*. This ability to dynamically interact with content on an as-needed basis, with no requirement to script these interactions, is one of the major strengths of this system.

While the questionnaire revealed that the audience responded positively to bullet highlighting, the presenter found Maestro's implementation useful in bringing attention to a *set* of bullet points all at once. In particular, Maestro implements a gradual fade-out of highlighted points. This enabled the presenter to sweep his hand across a range of points, to highlight them all at once. Note that this waving or sweeping gesture, for grouping objects, was also noted in the observational study as serving a similar purpose. This mass-highlighting of bullet points was not planned for, but became a welcome emergent feature of the system.

5.3.3 Discussion of the survey results and open-ended feedback

From the survey results, and open-ended feedback, we find that the audience responded quite well to gestures that interact with slide content such as highlighting bullet points and zooming into figures. The presenter was also enthusiastic about these features. These positive results are in spite of a general perception that the gesture recognizer has poor performance in recognizing these gestures. This strongly suggests that gesture-based presentation systems benefit from enabling content-centric gestures.

At the same time, the benefits of Maestro's *navigation* gestures are less clear; the audience responded poorly to scrollable slides, and the utility of the slide carousel was thrown into question (since there was never a need to use this feature during the two-week trial). Of course, the "next slide" and "previous slide" gestures were used quite frequently, and were perceived as being recognized with good accuracy; but, these commands are available on any wireless presentation remote. Since remotes are already both efficient and reliable, and because natural gestures typically arise only when the presenter is in the midst of explaining a slide, it is difficult to argue for a gesture-based alternative to issuing these two commands.

Finally, the survey results reinforce the challenge of providing feedback an affordances to the presenter without distracting the audience. With Maestro, we felt that subtle affordances could be provided to the presenter using small translucent cursors. Despite our best efforts, the majority of the audience reported that these cursors were distracting.

5.4 Side effects on presentation dynamics

While the ability to directly interact with content proved useful, Maestro's requirement that all interaction occur through gestures had a number of unintended side effects. These issues can be summarized as the *anchor problem*; the *field-of-view problem*; and, the introduction of a *no-fly zone.* We describe each in turn.

The anchor problem

One of the most visible effects of the system was that it tended to "anchor" the presenter next to the screen so he could control the presentation (e.g., advance slides). While this side effect was previously noted by Cao *et al.* in [5], this anchoring led to a number of unexpected outcomes, which we expand upon.

Maestro's placement of the staging area caused the presenter to locate himself just outside the edge of the screen. However, because the presenter must frequently face the screen to ensure that gestures are performed on their intended targets, this positioning encouraged the presenter to angle his body away from part of the class. This pivoting was not always corrected, leading the presenter to miss questions from students not in his field of view. Since the staging area was always incorporated into the left margin of the slides, it was always the same potion of the class whose questions were missed. In contrast, the location of the lectern (and, hence, laptop) in the corner of the room provided a clear view of the entire class. This finding strongly suggests the need to reexamine location-independent navigation gestures, or, at the very least, the ability to configure the system so it can be controlled from either side of the presentation screen.

The field-of-view problem

The tendency for the presenter to anchor himself next to the screen also made it difficult for the presenter to see all of the content being projected – what we term the *field-of-view problem.* After advancing to the next slide, the presenter would sometimes need to step back 4-5 feet from the screen to be able to see all of the slide's contents. From the audience's perspective, this behavior caused an obvious break from the presentation flow, and could be interpreted as the presenter being unprepared (when in fact the presenter simply needed to recall the points he wished to make). In contrast, when giving a presentation using a keyboard or remote control, the presenter was typically in a position to easily view each new slide in its entirety, whether it was on the laptop or projection screen. Glancing at a slide in these latter contexts is far less distracting since the presenter does not need to make a visible effort to look at the slide.

The no-fly zone problem

Finally, the design of the gesture recognition system also created a *no-fly zone* – a volume of space that the presenter could not enter without the risk of distracting the audience. Maestro was designed with the assumption that the presenter normally stands to the side of the projected content, only occasionally entering the projected content to selectively interact with elements in the slide. This is a safe assumption to make, since the presenter typically wishes the audience to be able to view the content without interference. However, after the first day of lecturing, the presenter found himself forgetting about the system and fully immersing himself in the act of lecturing. At times, he would wander in front of the projected content to address the class, gesticulating as he did so. This would lead to constant activity in the slides behind him, with bullet points automatically highlighting and un-highlighting as the presenter's hands unknowingly moved over these objects. Recall that Maestro instantly and automatically highlights bullet points whenever the hands are within their proximity (similar to the "mouse over" event in WIMP interfaces). This created an obvious distraction for the class. Recognizing this issue, the presenter consciously reduced his travel into and *through* this area. Accordingly, this *no-fly zone* served to further limit the presenter's movements.

5.5 Discussion

In this chapter we presented results from a prolonged real-world evaluation of Maestro. Our study suggests that gesture-based interaction leads to more interactive, but less efficient, presentations in comparison to PowerPoint. Additionally, the study found that Maestro's content-centric gestures were welcomed by both the presenter and the audience alike. However, the benefits of using gestures to navigate a presentation are less clear; especially when one considers that existing wireless remotes already enable efficient and reliable access to this functionality. Finally, Maestro's evaluation also illustrates some potential side effects of relying exclusively on gesture-based control of presentations. These effects include the *field-of-view*, *anchoring*, and *no-fly zone* problems.

The results of this work suggest several areas for future research. First, and foremost, multimodal interaction seems to hold great promise in this area. For navigating slides a wireless remote, or even a keyboard, might be the optimal solution. This traditional form of interaction is reliable and it leads to efficient presentations. On the other hand, gestures seem well suited for supporting rich, direct interaction with slide content. Creating a system that elegantly balances the multiple input modalities should result in a more optimal experience for both presenters and audience members.

Given the value we found for enhancing communication through content gestures, there is a need to more fully explore this design space. For example, gestures could be used to manipulate the parameters of a mathematical plot or simulation.

The benefits of such manipulations are well articulated by Douglas Zongker and David Salesin in their SLITHY presentation system [70]. SLITHY makes heavy use of parameterized diagrams and interactive objects using traditional input mechanisms. Extending this type of system to afford gesture-based control has yet to be explored.

Finally, while it is certainly important to evaluate Maestro based on feedback provided by both the audience and the presenter, the actual performance of Maestro's gesture recognizer must also be carefully evaluated in a controlled setting. Part II of this document, which begins in the next chapter, will report the results of this evaluation. In this latter section of the document, a principled gesture recognizer will be presented which is designed to replace the ad-hoc recognizer used thus far. These two recognizers will be directly compared in a controlled laboratory test, and the accuracy of both recognizers will be reported. It suffices to say, it is difficult to significantly outperform a finely tuned ad-hoc recognizer.

Part II

Gesture Recognition with Discrete Hidden Markov Models

Chapter 6

Discrete Hidden Markov Models for Modeling Gestures

In previous chapters, it was noted that Maestro's original gesture recognizer was "ad-hoc", and involved heuristic template matching. Various features of the hand trajectories (such as path length and center of mass) were compared to manually established range constraints. This simple approach allowed for rapid prototyping, and appeared to be rather good at spotting gestures. However, the ad-hoc approach is not generalizable and leaves much to be desired. In this chapter, we introduce a recognizer based on discrete hidden Markov models which is more representative of the state-of-the-art.

6.1 Introduction

The need to "fail-fast" when prototyping novel interfaces is well recognized in the field of human-computer interaction. In other words, it is important to be able to quickly identify and reject poor design choices in order to arrive at a final design. By using mock-ups, wizard of Oz user trials and other low-cost prototyping techniques, one can quickly arrive at a final design without investing too many resources in failed approaches. In support of rapid prototyping, Maestro was initially designed to use a collection of ad-hoc templates and heuristics for recognizing various gestures. While this approach proved to be an effective means for prototyping, it leaves much to be desired for a final system. In particular, the "ad-hoc" approach is not generalizable, requiring new heuristics to be developed for each new gesture that is to be recognized.

Upon completing the real-world evaluation of Maestro, we sought to develop a more principled and generalizable gesture-recognizer that is able to recognize Maestro's gesture language with similar or better accuracy as compared to the ad-hoc approach. In the literature, there are many approaches to recognizing hand gestures including dynamic time warping [10], time delay neural network [67], conditional density propagation (using particle filtering) [3], and various forms of hidden

Markov models [28, 65, 66, 48]. These approaches are well described in numerous survey papers regarding gesture-recognition [44, 38]. For Maestro, we elected to use a discrete hidden Markov model, since this method supports a relatively straightforward approach to spotting gestures embedded in longer motion sequences. In the remaining sections of this chapter, we review hidden Markov models, and present the DHMMs used to model gestures in Maestro.

6.2 Discrete hidden Markov models

Discrete hidden Markov models (DHMMs) are graphical models used for modeling sequential observations, such as the evolution of signals over time. These models were first described in the late 1960's and were almost immediately applied to the problem of speech recognition [47]. Since then, hidden Markov models have been widely applied to gesture-recognition [28, 66, 48, 65, 11]. An excellent description of these models is presented by Lawrence Rabiner in [47].

Formally, DHMMs adhere to the dynamic Bayesian network depicted in figure 6.1b, and consist of the following components:

- A set of N states $\mathbf{S} = \{s_1, s_2, \ldots, s_N\}$
- A discrete alphabet of M output symbols $\mathbf{\Sigma} = \{f_1, f_2, \ldots, f_M\}$
- A prior distribution $P(Q)$, where Q is a random variable over the set of initial states. Since there are finitely many states, the prior probabilities can be summarized by the stochastic vector $\pi \in \mathbb{R}^N$ whose i^{th} entry is defined as follows: $\pi_i = P(Q = s_i)$.
- A state transition distribution $P(Q_{t+1}|Q_t)$ where Q_t, and Q_{t+1} are latent random variables denoting the model's state at times t and $t+1$ respectively. Note that the transition distribution is conditioned only on the previous state Q_t. The resulting transition distributions can be summarized by the stochastic matrix $A = [a_{ij}]$, where $a_{ij} = P(Q_{t+1} = s_j | Q_t = s_i)$.
- A observation distribution $P(O_t|Q_t)$ where O_t is an observable random variable denoting the symbol from the alphabet that is generated by the model at time t. Note that the observation distribution is conditioned only on the current state Q_t. Each state has one conditional observation distribution, and the set of all observation distributions is summarized by the set $\mathbf{B} = \{b_i(j), i \in 1, 2, \ldots, N\}$, where $b_i(j) = P(O_t = f_j | Q_t = s_i)$
- Finally, the notation $\lambda = \{\pi, A, \mathbf{B}\}$ denotes the full set of parameters for any given DHMM.

It is also sometimes useful to interpret discrete hidden Markov models as being similar to *Moore* finite state machines (although a Mealy machine formulation is also possible). Like a Moore machine, a DHMM can be depicted graphically (as in figure 6.1a) where nodes depict the DHMM's states, and the arcs depict possible transitions from one state to the next. This should not be confused with the Bayesian network (figure 6.1b). In general, we use a state machine graph when discussing a DHMM's topology, and the Bayesian network graph when reasoning about conditional independence. As with a Moore machine, DHMMs output one symbol from the discrete output alphabet upon arriving at each state. Again, the output at time t depends only on the Machine's state at that time (and not on the transition taken to arrive at the state). However, unlike a Moore machine, a state's output is selected at random from the state-specific output distribution. Moreover, the DHMM's state transitions are also governed by a random process, rather than by an input sequence.

Regardless of which description one wishes to use when picturing hidden Markov models, it is important to keep in mind the properties that give rise to their namesake; first, the state sequence evolves according to a 1^{st}-order Markov process, with each state transition depending only on the previous state; secondly, the state sequence is always *hidden* from the observer, who only has knowledge of the output sequence; the state sequence must be inferred from the observations.

6.2.1 Filtering, decoding, and parameter estimation

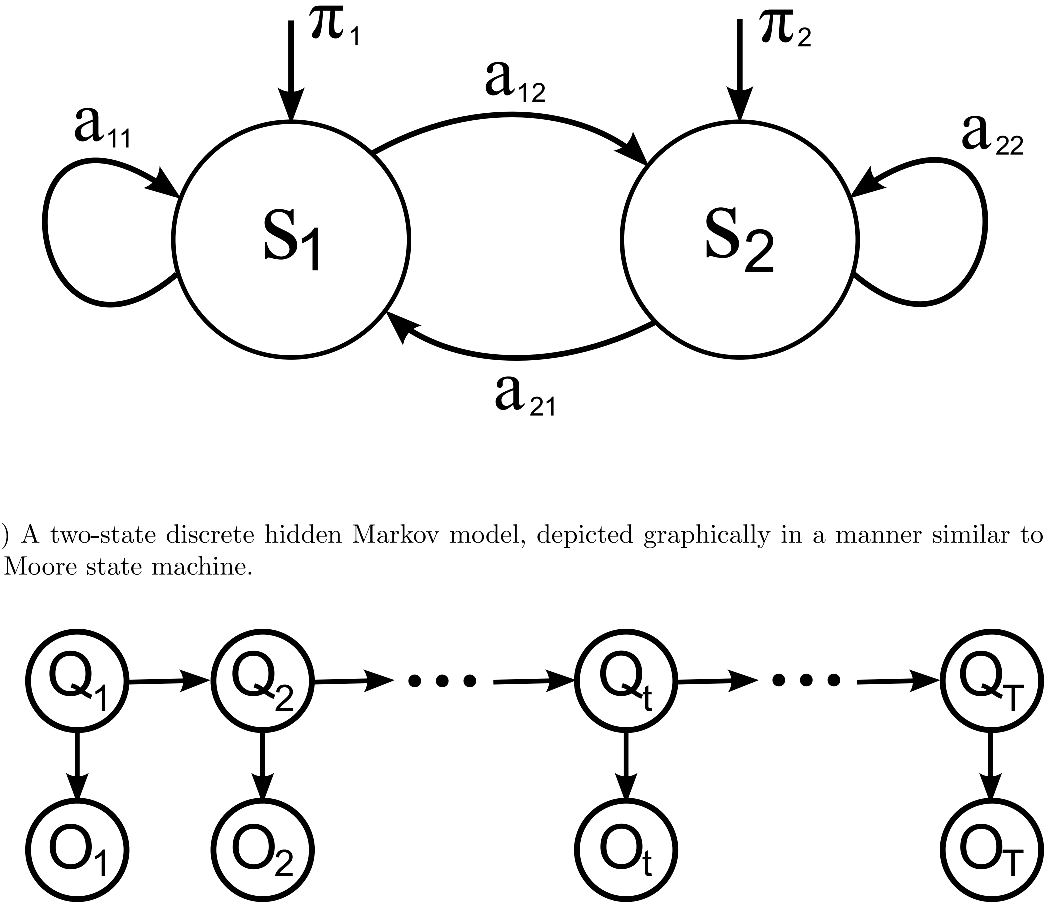
Rabiner's tutorial on hidden Markov models [47] lists three basic problems for HMMs whose solutions have many practical applications. These problems include filtering, decoding and parameter estimation. Again, we present only a brief overview of these problems, deferring a more thorough treatment to Rabiner's tutorial and other works of interest [32].

Filtering

Filtering involves determining the probability of the observation sequence $O_{1:T} = O_1\ O_2\ \ldots\ O_t\ \ldots\ O_T$, given the model parameters λ, i.e., $P(O_{1:T}|\lambda)$. For a given state sequence $Q_{1:T} = Q_1\ Q_2\ \ldots\ Q_t\ \ldots\ Q_T$ then:

$$\begin{aligned}
&P(O_{1:T}, Q_{1:T}|\lambda) = \\
&\quad P(Q_1|\lambda)P(O_1|Q_1,\lambda)P(Q_2|Q_1,\lambda)P(O_2|Q_2,\lambda)\ldots \\
&\quad P(Q_T|Q_{T-1},\lambda)P(O_T|Q_T,\lambda) \qquad (6.1)
\end{aligned}$$

Thus $P(O_{1:T}|\lambda)$ can be obtained through marginalization, by summing out all possible state sequences. However, this computation is intractable, since there are N^T



(a) A two-state discrete hidden Markov model, depicted graphically in a manner similar to a Moore state machine.

(b) A Bayesian network for a hidden Markov model.

Figure 6.1: A two-state DHMM viewed both as a stochastic state machine, and as a dynamic Bayesian network. The state machine graph 6.1a is used when describing the model topology, while the Bayesian network graph 6.1b is used when reasoning about conditional independence.

possible state sequences for an observation sequence of length T (and where N is the number of states). Thankfully, there exists the FORWARD-ALGORITHM for computing $P(O_{1:T}|\lambda)$ in $O(N^2T)$ computations. The algorithm works by passing a forward message along a state-time lattice. The forward message $\alpha_t(i) = P(O_{1:t}, Q_t = s_i|\lambda)$ represents the probability of observing the first t observations, and ending the state sequence in state s_t. Following Rabiner's tutorial, $\alpha_t(i)$ can be computed inductively as follows:

1. Base case, where $t = 1$.

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad i \in \{1, 2, \ldots, N\} \tag{6.2}$$

2. Inductive step, $1 < t < T$.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad j \in \{1, 2, \ldots, N\} \tag{6.3}$$

3. Termination

$$P(O_{1:T}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{6.4}$$

Decoding

Another task common with DHMMs is decoding. Decoding addresses the problem of determining the state-sequence $q^*_{1:T}$ which "best" explains the observation sequence. Typically, this means finding the state sequence which, together with the observations, maximizes the likelihood function:

$$q^*_{1:t} = \text{argmax}_{q_{1:T}} P(q_{1:T}, O_{1:T}|\lambda) \tag{6.5}$$

The solution to this problem can be computed efficiently using the Viterbi algorithm, which is a dynamic programming algorithm that again relies on the state-time lattice. In fact, the Viterbi algorithm is very similar to the aforementioned FORWARD-ALGORITHM, requiring only a minor modification and some extra bookkeeping to allow for backtracking.

Parameter estimation

The final task common when using DHMMs is that of learning the model parameters, λ, which best account for a given sequence of observations (and a given model topology). Using a maximum likelihood approach, λ can be estimated as follows:

$$\lambda_{ML} = \text{argmax}_\lambda P(O_{1:T}|\lambda) \quad (6.6)$$

Unfortunately, there is no known closed-form solution to this optimization problem. However, given an initial estimate λ_i of the model parameters, the Baum-Welch reestimation procedure allows the computation of λ'_i whose likelihood is greater than or equal to that of λ_i. By iteratively applying this reestimation procedure, the model parameters are moved towards a local maximum of the likelihood function until convergence. As usual, repeating the procedure with different initial conditions may lead to other local maxima, and may help broaden the search.

6.3 Model topologies for gestures

The general definition of a DHMM from Section 6.2 allows state transitions to occur between any pair of states. Such DHMMs are known as "ergodic", and have a fully connected state topology (i.e., a full state transition matrix A). However, in gesture recognition (and also in speech recognition), it is useful to consider other topologies; specifically, it is common to use a left-right topology [47, 28, 66, 48] where state s_i is connected to state s_j only if $j \geq i$. This results in an upper-triangular transition matrix A. Left-right models are ideal for modeling gestures and spoken words, both of which can be thought of as steadily progressing through a series of states. For instance, gestures often have a clear beginning, middle and end. It is also common to limit the number of states that can be skipped when moving from one state to the next (a so-called "constrained jump" model). Here, state s_i is connected to state s_j only if $0 \leq j - i \leq \Delta$.

In Maestro, all gestures are modeled using a 4-state constrained jump DHMM where $\Delta = 1$. This topology is depicted in figure 6.3. Note that the model includes a 5th non-emitting final state. This state is only entered after observing the final observation O_T. The use of a final state is quite common, and forces finite observation sequences to align with the full model. Conceptually, one can think of all finite observation sequences as being terminated by an "end of sequence" observation, which can only be generated by the final state.

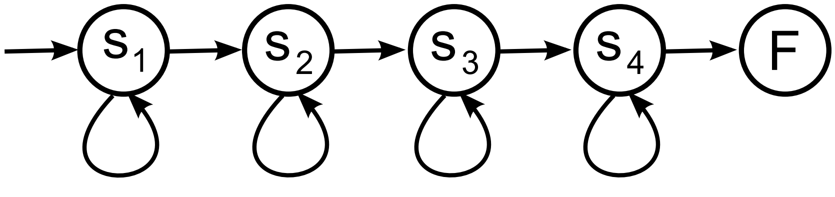


Figure 6.2: Topology of the four-state constrained jump DHMM used for modeling gestures in Maestro. The 5th state "F" is a final non-emitting state.

6.4 Feature extraction for gestures

In the previous section, we presented the topology of the DHMMs used by Maestro to model gestures. In this section, we describe how Maestro's computer-vision system generates the discrete observations required by the discrete hidden Markov model.

Maestro's computer vision system reports on the positions of both the red and blue gloves at a rate of 15 times per second. Although both gloves occupy a sizable area in every camera frame, the position of each glove is summarized by a single point in space; specifically, the rightmost point on the glove's contour (which is similar to the method used in [15]). Let $\vec{R}_t = [R_t^{(x)}, R_t^{(y)}]^T$, and $\vec{B}_t = [B_t^{(x)}, B_t^{(y)}]^T$ be the positions of the red and blue gloves respectively at time t. Together $\vec{R}_t$ and $\vec{B}_t$ can be used as basic features for a *continuous* hidden Markov model. However, the use of these features is ill-advised. To illustrate this point, consider Maestro's "undo" gesture (figure 4.1h), which is performed in the following four stages:

1. start with one hand at rest
2. move to the right with positive horizontal velocity (and zero vertical velocity)
3. move to the left with negative horizontal velocity
4. end in a state of rest upon returning to the initial position

This gesture seems well suited to be modeled with a 4-state HMM - one state for each of the aforementioned steps. However, when using positions $\vec{R}_t$ and $\vec{B}_t$ as features, the second and third states of the HMM pose a modeling challenge: While in the second state, the x-coordinate of the hand-position is monotonically increasing with time; and, when in the third state, it is monotonically decreasing. However, an HMM's conditional observation distributions $P(O_t|Q_t)$ cannot take into account the passage of time when assigning probabilities to observations; there is an implicit assumption that the conditional observation distributions do not change with time.

To combat this issue, numerous researchers have suggested using a measure known as "direction" or "turning angle" as a more stable feature [28, 41, 8]. Consider the pair of sequential observations $\vec{R}_{t-1}$, $\vec{R}_t$. The turning angle $\theta_t^{(R)}$ is defined as the angular component of the finite difference $\vec{R}_t - \vec{R}_{t-1}$, when expressed in polar coordinates:

$$\theta_t^{(R)} = \text{atan2}\left(R_t^{(y)} - R_{t-1}^{(y)},\ R_t^{(x)} - R_{t-1}^{(x)}\right) \tag{6.7}$$

Here, the function atan2 (y, x) is a variant of the arc tangent function which takes into account the quadrant in which the point (x, y) lies. This function is very useful for converting between cartesian and polar coordinates, was first introduced in the FORTRAN programming language, and has since become a standard math operator in many other programming languages (e.g., C, Java, Matlab) [60].

The turning angle $\theta_t^{(B)}$ of the blue glove is defined similarly. Initially, both $\theta_1^{(R)}$ and $\theta_1^{(B)}$ are undefined since there are no previous observations from which to compute the finite difference. Additionally, the red and blue gloves may not be present at all times. Suppose that the red glove is not detected at time t, then neither $\theta_t^{(R)}$ nor $\theta_{t+1}^{(R)}$ are defined.

Using the turning angle feature, it becomes much easier to accurately model linear gestures. Continuing the "undo" gesture example, the second state of the model would have a conditional observation distribution with one mode centered around $\theta_t \approx 0°$. This would represent the hand moving to the right. Similarly, the third state would have a conditional observation distribution with one mode centered around $\theta_t \approx 180°$. In general, turning angle is an excellent feature for modeling piecewise linear gestures. Of course, the turning angle must be discretized before this feature can be used in a discrete hidden Markov model. This discretization is described in the next section.

Discrete turning angle

If the turning angle $\theta_t^{(R)}$ is constrained to the range $[0°, 360°)$, the feature can be discretized by simply dividing the range into equal sized bins (e.g., 12 bins, each accounting for 30°). The discretization $\Theta_t^{(R)}$ for $\theta_t^{(R)}$ is simply the index of the bin to which the continuous turning angle is assigned (figure 6.3a). This approach to discretizing the turning angle is very straightforward, and has been used previously in [28].

Additionally, we noted earlier that the turning angle is undefined in the cases where the glove is not detected. Moreover, the measure itself becomes unstable when the hands are at, or are near, rest. To resolve these issues in the discretization, we simply add one bin for each of these cases. This requires thresholding the finite difference $||R_t - R_{t-1}||_2$ in order to determine when the glove is considered to be "near rest".

Regional context

While the turning angle feature captures the motion of the hands, it provides no information regarding the spatial context in which the motion occurs. As mentioned in Chapter 4, many of Maestro's gestures are contextualized by particular targets or regions of interest (ROIs) such as bullet-points or figures. Consequently, we compute a discrete feature which captures this spatial information. At each instant, the hands can find themselves in one of three spatial contexts known as "zones":

- ZONE 1: "Inside" the region of interest

- Zone 2: “Near” the region of interest (i.e., *within* ϵ pixels from the region, either horizontally or vertically, as depicted in figure 6.3b).
- Zone 3: “Far” from the region of interest (i.e., *more* than ϵ pixels from the region, either horizontally or vertically).

The feature $Z_t^{(R)}$ encodes the zone in which the red glove is found at time t. If the red glove is not detected at time t, then $Z_t^{(R)}$ takes on a 4th value indicating that the hand is absent. $Z_t^{(B)}$ is defined similarly, but for the blue glove.

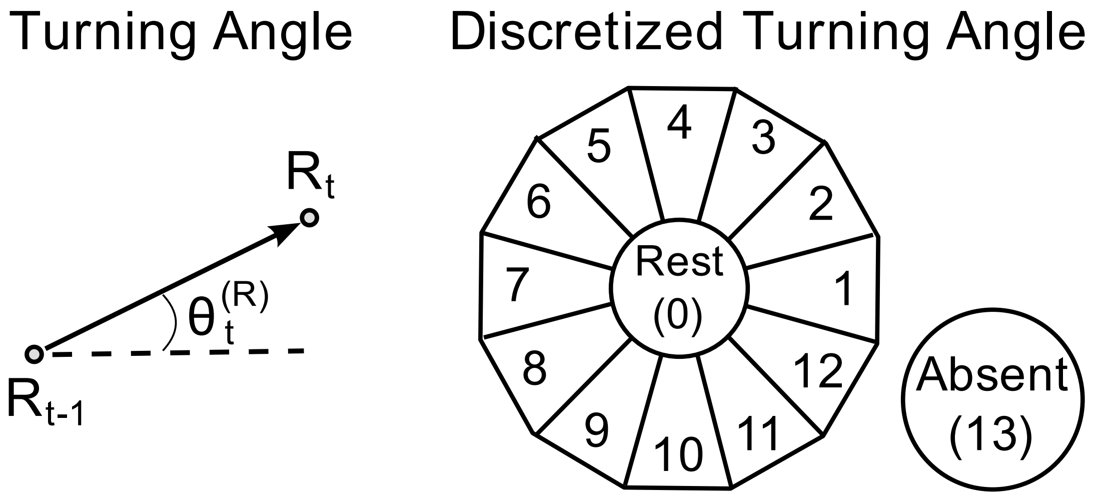
Spatial relation between hands

As mentioned in the previous section, the turning angle captures the motion of the hands, but not their configuration with respect to one-another. For example, it provides no indication of whether the hands are together, or if they are collinear along a column or row of the display, etc. To capture this information, we introduce the spatial relation feature σ_t which is similar to the turning angle features but is computed using the difference vector $\vec{R}_t - \vec{B}_t$ rather than $\vec{R}_{t-1} - \vec{R}_t$. In this sense, σ_t encodes the direction of the vector pointing from the blue glove towards the red glove. Importantly, σ_t becomes unstable when $||\vec{R}_t - \vec{B}_t||_2$ is small. These short vectors indicate that the hands are “together”.

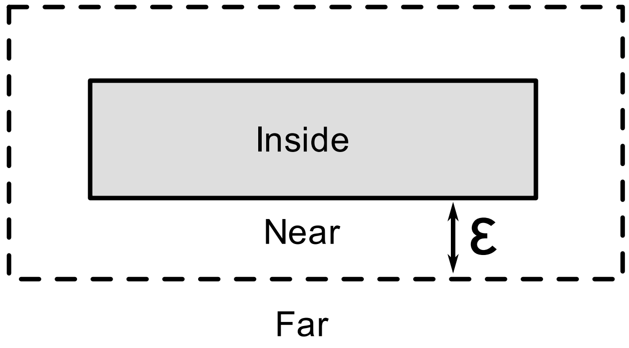
6.4.1 Maestro’s conditional observation distributions

At this point, the observations are not discrete integers but are instead discrete feature vectors $\vec{f} = [\Theta_t^{(R)}, Z_t^{(R)}, \Theta_t^{(B)}, Z_t^{(B)}, \sigma_t]^T$. Given our current definition of a simple DHMM, it is unclear how to use feature vectors as observations. One possible solution is to use vector quantization or a “code book” to convert feature vectors into unique integers [66, 45, 47]. With a finite number of possible feature vectors, a simple code book can be constructed by enumerating the vectors, and using a vector’s index in the enumeration as its codeword. However, this simple approach is dreadfully wasteful given the number of possible feature vectors. Consider that there are 14 possible values for each of the turning angle features $\Theta_t^{(R)}$ and $\Theta_t^{(B)}$, 4 possible values for each of the zone features $Z_t^{(R)}$ and $Z_t^{(B)}$, and 14 more possibilities for the spatial relationship feature σ_t. Together, this makes for $14^3 \times 4^2 = 43,904$ possible feature vectors. An HMM with 4 states would then require more than 175,000 parameters. Learning such a model would require an immense amount of training data in order to acquire good approximations for each of these parameters.

One way to resolve this issue is to assume that each of the feature vector’s components are conditionally independent given the HMM state (as in figure 6.4a). This reduces the number of parameters to 50 for each state – an immense savings! However, such a factorization is unwarranted since the various components of $\vec{f_t}$ are not conditionally independent. For example, if the feature $Z_t^{(R)}$ indicates that



(a) The turning angle feature, and it's discretization.



(b) The spatial zone feature.

Figure 6.3: The discrete turning angle and zone features used for modeling gestures.

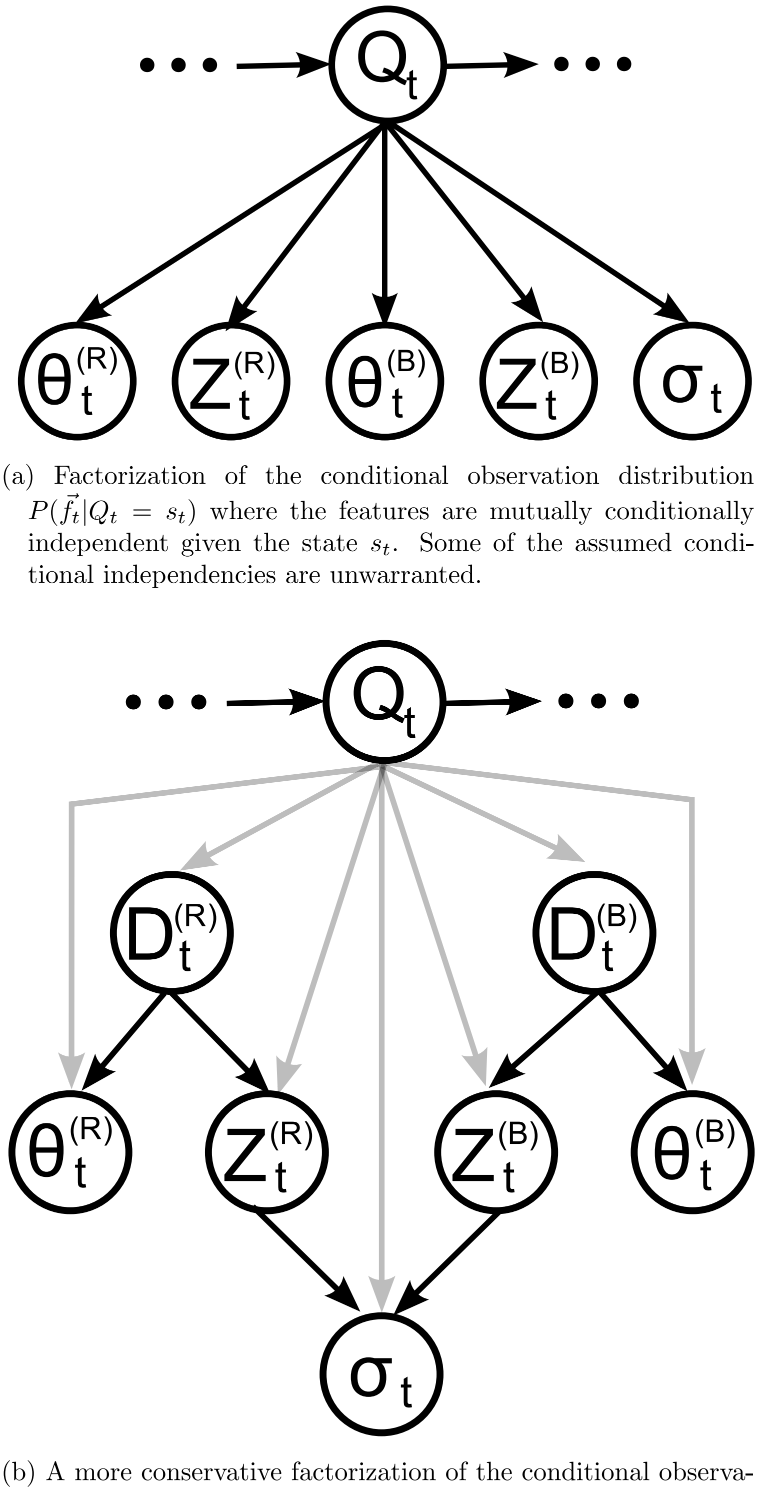
the red glove was not detected, then it follows that $\theta_t^{(R)}$ will take on a similar "not detected" value. Similarly, if $Z_t^{(R)}$ indicates that the red glove is "inside" a region of interest, while $Z_t^{(B)}$ indicates that the blue glove is "far" from the region of interest, then we should not expect the feature σ_t to take on a value indicating that the hands are close together.

In Maestro, the conditional observation distributions $P(\vec{f_t}|Q_t)$ are factored according to the Bayesian network depicted in figure 6.4b. This factorizing does not require one to assume inappropriate conditional independencies, but requires the addition of two new binary random variables $D_t^{(R)}$ and $D_t^{(B)}$. These random variables indicate if the red and blue gloves have been detected at time t. As a result, the observation distributions $P(\vec{f_t}|Q_t)$ each require only 300 parameters, and are factored as follows:

$$\begin{aligned}
&P([D_t^{(R)}, \Theta_t^{(R)}, Z_t^{(R)}, D_t^{(B)}, \Theta_t^{(B)}, Z_t^{(B)}, \sigma_t]^T|Q_t) = \\
&\quad P(D_t^{(R)}|Q_t)P(\Theta_t^{(R)}|D_t^{(R)}, Q_t)P(Z_t^{(R)}|D_t^{(R)}, Q_t) \times \\
&\quad P(D_t^{(B)}|Q_t)P(\Theta_t^{(B)}|D_t^{(B)}, Q_t)P(Z_t^{(B)}|D_t^{(B)}, Q_t) \times \\
&\quad P(\sigma_t|Z_t^{(R)}, Z_t^{(B)}, Q_t) \qquad (6.8)
\end{aligned}$$

6.5 Discussion

This chapter provided a brief background into discrete hidden Markov models, and presented the specific models and features used for representing gestures in Maestro. Importantly, Maestro's DHMMs make use of a factored observation distribution which reduces the number of parameters needed to model each gesture. Additionally, these observation distributions begin to capture many of the non-accidental motions which Maestro's gestures were originally designed to incorporate (as described in Chapter 4). For example, axis-aligned hand motion is well-captured by an observation distribution where the turning angle features are explained by compact modes centered around the directions corresponding with the axes (i.e.,0°, 90°, 180° or 270°). Similarly, many non-accidental aspects of bimanual interaction are also well represented in this way (e.g, cases where the hands are moving in parallel, or collinearly, induce compact modes in the distributions over the spatial relation feature σ_t). Each of these compact observation distributions corresponds to a highly specific motion event. The sequence of such motion events is then dictated by the topology of the encompassing DHMM. As a result, each gesture model accounts for only a very particular set of hand motion. Nonetheless, differentiating gestures from other hand motions is a difficult and complex task. In the next chapter, we describe how Maestro uses the aforementioned DHMMs to spot meaningful gestures that are embedded in sequences of unconstrained hand motion.



(a) Factorization of the conditional observation distribution $P(\vec{f_t}|Q_t = s_t)$ where the features are mutually conditionally independent given the state s_t. Some of the assumed conditional independencies are unwarranted.

(b) A more conservative factorization of the conditional observation distribution.

Figure 6.4: Possible factorizations of the conditional observation distributions $P(\vec{f_t}|Q_t = s_t)$.

Chapter 7

Gesture Spotting

The previous chapter described the discrete hidden Markov models used by Maestro to model gestures. This chapter describes the application of these models to the problem of spotting gestures in continuous motion sequences. In order to highlight the challenges, and to motivate our approach, we begin with the far simpler problem of isolated gesture recognition. This problem and its solution are then contrasted with the more challenging problem of gesture spotting.

7.1 Isolated gesture recognition with DHMMs

In isolated gesture recognition, the observation sequence is assumed to correspond to the complete performance of exactly one gesture. This scenario is frequently applied when recognizing written gestures, where pen contact with the writing surface provides a clear indication of where each gesture begins and ends. Isolated gesture recognition is closely related to isolate word recognition in spoken language, where silences between spoken words are used to locate word boundaries [54]. In both cases, the recognition task is simply a closed-world classification problem. For gesture recognition, each observation sequence is attributed to a specific gesture in the gesture vocabulary $\mathbf{G} = \{G_1, G_2, \ldots G_N\}$. Suppose that the observation sequence $O_{1:T}$ is to be classified. If the gesture prior probabilities $P(G_i)$ are known, then classification can proceed using the maximum a posteriori (MAP) decision rule:

$$G_{MAP} = \text{argmax}_{G_i \in \mathbf{G}} P(O_{1:T}|G_i)P(G_i) \tag{7.1}$$

If the prior probabilities are unknown, then a maximum likelihood (ML) decision rule can be used instead. This decision rule effectively assigns equal prior probability to each of the gestures:

$$G_{ML} = \text{argmax}_{G_i \in \mathbf{G}} P(O_{1:T}|G_i) \tag{7.2}$$

In both cases, the likelihood $P(O_{1:T}|G_i)$ is given by filtering using the corresponding hidden Markov model with parameters λ_i. Recall that the Forward-Algorithm provides an efficient method for calculating $P(O_{1:T}|\lambda_i)$.

7.2 Isolated gesture recognition with "non-gesture" rejection

In isolated gesture recognition, the onus of isolating the gesture instances falls on the user of the system. Again, in the pen-based example described in the previous section, the user indicates their intention to form a gesture by contacting the pen with the writing surface. The gesture is then ended by releasing contact. In this scenario, errors in segmentation are quite common. For example, the user may initiate a gesture but abort the gesture mid-sequence. Alternatively, the user may error when forming the gesture, or may initiate the gesture recognition process entirely by mistake. These situations lead to misclassifications or false detections. For this reason, it is desirable to be able to detect "non-gesture" sequences. One possible approach is to use the MAP or ML criteria to find the best gesture model λ_*, but to establish a likelihood threshold K in order to reject non-gestures:

$$\text{Reject} \;\; O_{1:T} \;\; \text{if} \;\; P(O_{1:T}|\lambda_*) < K \tag{7.3}$$

In this context, the likelihood $P(O_{1:T}|\lambda_*)$ is treated as a confidence measure; if the likelihood is too low, the classification is considered untrustworthy. The problem with this approach is that the likelihood tends to be highly dependent on the sequence length T, decreasing quickly as T increases [28, 45, 49]. Consequently, it is very difficult to establish a single likelihood threshold K that gives good performance across the full range of temporal variability associated with the gesture.

While likelihood alone is a poor confidence measure, the idea of using a confidence measure to reject non-gesture sequences is reasonable, and has been widely used in the related field of speech recognition [49, 45, 63, 69]. As a result, confidence measures have seen a great deal of attention in this literature (see [19] for a good review). Unfortunately, the same cannot be said about gesture recognition. One possible reason for this discrepancy is that, in speech recognition, language models, grammar and other high-level constraints can be used as a basis for developing accurate confidence measures. With the exception of systems that recognize sign language, gesture recognizers do not have this luxury. The remainder of this section explores some confidence measures that are applicable to the problem of gesture recognition.

One simple approach to transforming likelihood into a confidence measure is to normalize the log-likelihood by dividing by the sequence length:

$$\log \overline{P(O_{1:T}|G_i)} = \frac{\log P\left(O_{1:T}|G_i\right)}{T} \tag{7.4}$$

where,

$$\overline{P(O_{1:T}|G_i)} = \left(P\left(O_{1:T}|G_i\right)\right)^{1/T} \tag{7.5}$$

This result can be interpreted as the average probability of each individual observation in the sequence [45, 63]. As before, non-gesture sequences can be rejected by applying a threshold to this confidence measure. While this approach may account for sequence length, it remains problematic. Importantly, it treats all observations equally, tending to overvalue generic observations, while undervaluing observations that may be particularly informative. As an example, a sequence may be assigned high confidence only because it exhibits properties that are common to other gestures. In this case, the sequence's specific classification is uncertain – yet this uncertainty is not reflected by the confidence measure.

In contrast to the likelihood, the ideal confidence measure is simply the posterior probability of the gesture given the observations [4, 19, 21], i.e., $P(G_i|O_{1:T})$. This is an absolute measure of confidence ranging from 0 to 1. It is also context independent, easy to interpret, and can be recovered using Bayes theorem:

$$P(G_i|O_{1:T}) = \frac{P(O_{1:T}|G_i)}{P(O_{1:T})}P(G_i) \tag{7.6}$$

It is also quite common to use only the quotient in the above equation, which has been referred to as "normalized likelihood" [21]. Note that the denominator, $P(O_{1:T})$, is the prior probability of the observation sequence. Unfortunately, $P(O_{1:T})$ is very difficult to model directly [4, 19].

An alternative approach is to compare the likelihood of the hypothesized classification G_h to the likelihood of a competing model G_0. This amounts to a likelihood ratio statistical test, where G_0 is the null hypothesis. In this case, the null hypothesis is rejected if:

$$\frac{P(O_{1:T}|G_h)}{P(O_{1:T}|G_0)} > K_c \tag{7.7}$$

where K_c is the critical value of the test, while the ratio is itself a measure of confidence. The argument for using this test is that if there is ambiguity in assessing which model best fits the observations, the classification result is likely to be unreliable. The question is then which model should be used for G_0. One common approach is to develop an "anti-model", $\overline{G}_i$, for each gesture G_i. The anti-model allows one to approximate the likelihood of the sequence given that the observations *do not correspond with the gesture* (i.e., $P(O_{1:T}|\overline{G}_i)$. In this case, the likelihood ratio is related to the odds-ratio by the following equation:

$$\frac{P(G_i|O_{1:T})}{P(\overline{G}_i|O_{1:T})} = \frac{P(O_{1:T}|G_i)}{P(O_{1:T}|\overline{G}_i)}\ \frac{P(G_i)}{P(\overline{G}_i)} \tag{7.8}$$

The odds-ratio is useful because it represents the odds that the observation sequence was generated by the gesture model G_i.

Unfortunately, developing an accurate "anti-model" is about as difficult as developing a model for the observation prior $P(O_{1:T})$; after all the two are related by:

$$P(O_{1:T}|\overline{G}_i) = \frac{P(O_{1:T}) - P(O_{1:T}|G_i)P(G_i)}{1 - P(G_i)} \tag{7.9}$$

Notice that all terms on the right-hand side of the equation are easily computed except for the observation prior.

In practice, speech recognition systems often construct anti-models by modeling one or a few words that are acoustically similar to the target keyword [4]. These words are more likely to result in classification errors. This approach is similar to "cohort normalization" in speaker verification systems, where the anti-models are constructed from a cohort (i.e., a collection) of likely impostors [50, 16].

7.3 Continuous gesture recognition ("gesture spotting")

In continuous gesture recognition, it is not known where gestures start or end within the observation sequence. In this environment, gestures must be both isolated (i.e., segmented) and recognized simultaneously. This problem is known as "gesture spotting", and is analogous to keyword spotting in speech recognition systems. One simple approach is to explicitly test all possible subsequences of the observation sequence. Subsequences which do not correctly isolate gestures can be rejected using the confidence measure thresholding technique presented in the previous section [63]. Since an exhaustive search over all possible subsequences is computationally expensive, it is common to constrain the search to a fixed-length sliding window [22]. In either case, many subsequences are tested, but only a few will correspond to gestures. Consequently, the approach demands an accurate decision rule for rejecting non-gesture segments.

In contrast to the sliding window approach, a much simpler and more efficient approach is possible if one has access to a model for non-gestures. In other words, we need a model for the distribution $P(O_{1:T}|BG)$, which is the likelihood of the observation sequence given that the observations were generated by some background process rather than by the performance of *any* gesture. Here the background-model should not be confused with the gesture-specific anti-models of the previous section.

In gesture or keyword spotting applications it is common to model $P(O_{1:T}|BG)$ directly. If the background model is a hidden Markov model with parameters λ_{BG}, then it is referred to as a "garbage" or "filler" model [64, 11]. Such models "close the world" since all segments of the input sequence can be explained either by a gesture performance or by the background process. Consequently, a larger composite DHMM (figure 7.1) can be devised for the entire observation sequence [62, 11, 66, 28]. In this case, time-synchronous Viterbi decoding is often used to establish the most likely state sequence through this DHMM for a given observation sequence. This approach implicitly segments the observations into gesture and background subsequences. Gestures are spotted when the Viterbi path passes from beginning to end through a gesture model. Consequently, this larger DHMM is referred to as the "gesture spotting network" [28].

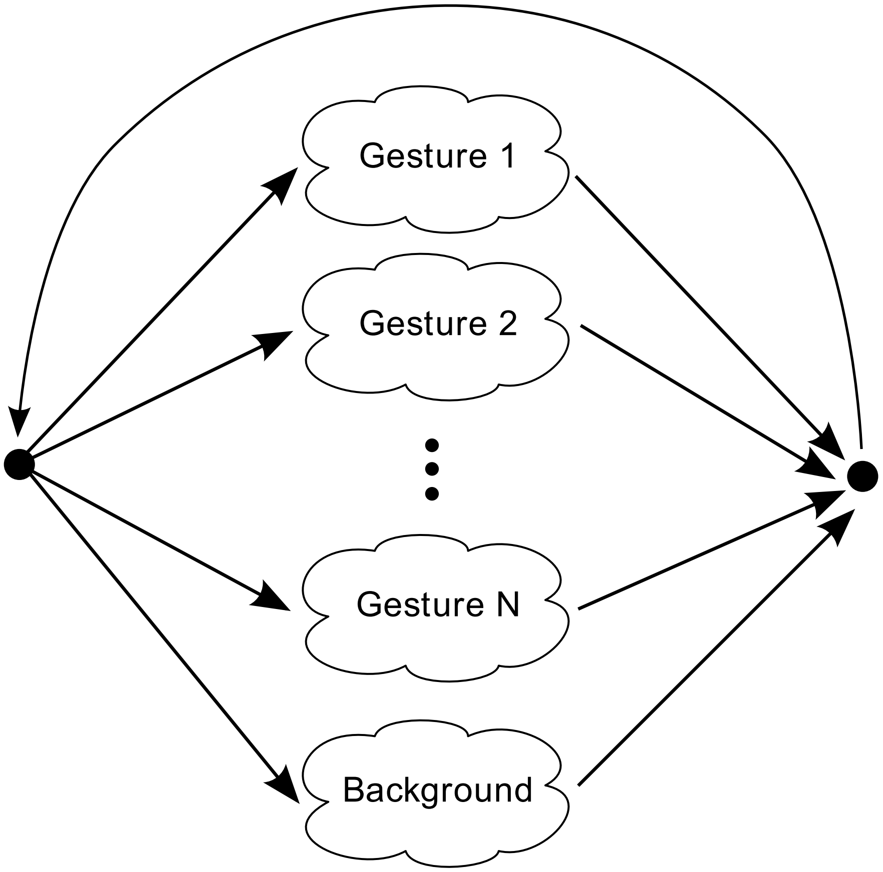


Figure 7.1: A generic gesture spotting network in which gestures and the background model are connected in parallel. The black dots represent non-emitting states (also known as "null states").

Unfortunately, it is quite difficult to devise a good background model for the same reasons it is difficult to generate anti-models or models of the observation prior: Non-gesture motion comes in "all shapes and sizes", and training such a model would involve an immense amount of training data. One way to mitigate the problem is to factor the background model by directly modeling certain specific types of non-gesture motion. A "catch-all" model can then be used to account for the remaining possibilities. For example, when applied to keyword spotting in speech recognition, specific garbage models typically represent silences, pops, microphone hisses, and transmission noise [64]. The remaining non-keyword sequences are likely to be words that are not found in the vocabulary. Such out-of-vocabulary words can then be modeled by a process which simply explains word formation by a random sequencing of phones (or triphones), where the probabilities associated with each phone (or triphone) are learned from a corpus of non-keyword speech samples [69]. Here, the explicit modeling of *common* non-keyword sequences (e.g. silences) ensures that a large percentage of the observations will be accounted for by dedicated models; while the factorization itself provides clues as to how to better model the entire background process.

Finally, the Viterbi approach suffers from one important drawback: it provides only the single most likely path through the gesture spotting network. There may be other possible paths with similar likelihoods, but these are not reported. In most cases, this is not a problem since the other high-likelihood candidates represent only minor variations to the Viterbi path; however, there may be cases where other high-likelihood paths pass through a different gesture, or through the background portion of the model. As with isolated gesture recognition, cases that do not result in a clearly superior solution suggest that the results may not be entirely reliable. Again, a confidence measure can be applied to detect these cases, and can be used to reject false positives. This is done in a post-processing step after the candidate gesture has been spotted.

7.4 Gesture spotting in Maestro

Since Maestro must be able to spot gestures in continuous motion sequences, it uses the Viterbi approach described in the previous section. Maestro's gesture spotting network (figure 7.2) includes one 4-state left-right DHMM for each gesture in the gesture vocabulary (as described in Chapter 6). For factoring the garbage model, Maestro uses a one-state "silence" model to account for sequences in which neither hand is detected. Maestro also uses the catch-all model suggested by Lee *et al.* in [28]. Lee's catch-all model is based on the observation that each state of each gesture model represents a known "substroke" or "sub-pattern" [1]. Examples of substrokes include periods of rest, or periods of linear motion in a particular direction, etc. Substrokes are comparable to phones in speech recognition. Consequently, the catch-all model simply explains all hand motion by an arbitrary ordering of substrokes. It is constructed by combining all states from all gesture models into one large ergodic (fully connected) DHMM. Since the duration of the substrokes is important, the catch-all model retains the original self-transition probabilities of all states. The remaining probability mass is evenly distributed among the remaining inter-state transitions. In addition to including these substrokes, we found it useful to include one state with a uniform observation distribution. This additional state

[1] Although Lee uses the terminology "threshold model" rather than "catch-all" model

ensures that novel sub-patterns are not assigned a zero probability by the catch-all model.

One important note is that the Viterbi gesture spotting approach has an unintended consequence when used for real-time gesture recognition. Specifically, the Viterbi path will not leave a state until the system is presented with evidence suggesting that it do so. This behavior is by design, and is exactly what a proper decoder should do. However, it means that a completed gesture will not be recognized until the user begins performing the next gesture (or some other non-gesture motion). This can cause what can only be described as a deadlock between the user and the system; the user is waiting for acknowledgement before beginning the next gesture, but the system cannot acknowledge the current gesture until the user moves on. For example, suppose the final state of the "next slide" gesture represents a period of rest. Upon performing the gesture, users tend to remain in the rest state until the gesture is recognized. Without any system acknowledgement, the user will eventually assume the gesture was missed, and will try again. The gesture will be recognized as soon as the user repositions to make his or her second attempt. A response at this time is quite unexpected and confusing.

One possible solution to this problem is to recognize a gesture as soon as the Viterbi path enters the gesture's final state rather than waiting for the path to exit; however, this would certainly be premature. Alternatively, one could define a maximum length of time that the Viterbi path can spend in a gesture's final state; however, this would effectively condition a state transition on the time spent in the state. The resulting model would no longer be an HMM. Maestro's solution is to acknowledge a gesture if the following two conditions hold:

- The Viterbi path ends in a gesture's final state.
- The Viterbi path, extended to include a transition out of the gesture's final state, remains the most likely path. Note that this extra transition leads to a non-emitting state (which does not consume an observation), and is thus a valid state sequence for the observations.

Using this strategy, gestures are recognized as soon as possible. However, if too little time is alloted to the gesture's final sate, the Viterbi decoder may prefer an alternative path leading through one of the garbage models.

As a final safeguard, once a potential gesture is isolated from the input sequence, it is subjected to a confidence measure threshold as described in the previous section. This is done is a post-processing step, after performing Viterbi decoding. The confidence measure thresholding can help in rejecting sequences with insufficient duration, as well as those that otherwise yield low confidence. For a confidence measure, Maestro uses the gesture instance's posterior probability, which is computed using the entire gesture spotting network to approximate the observation prior $P(O_{1:T})$ using the FORWARD-ALGORITHM described in Section 6.2.1.

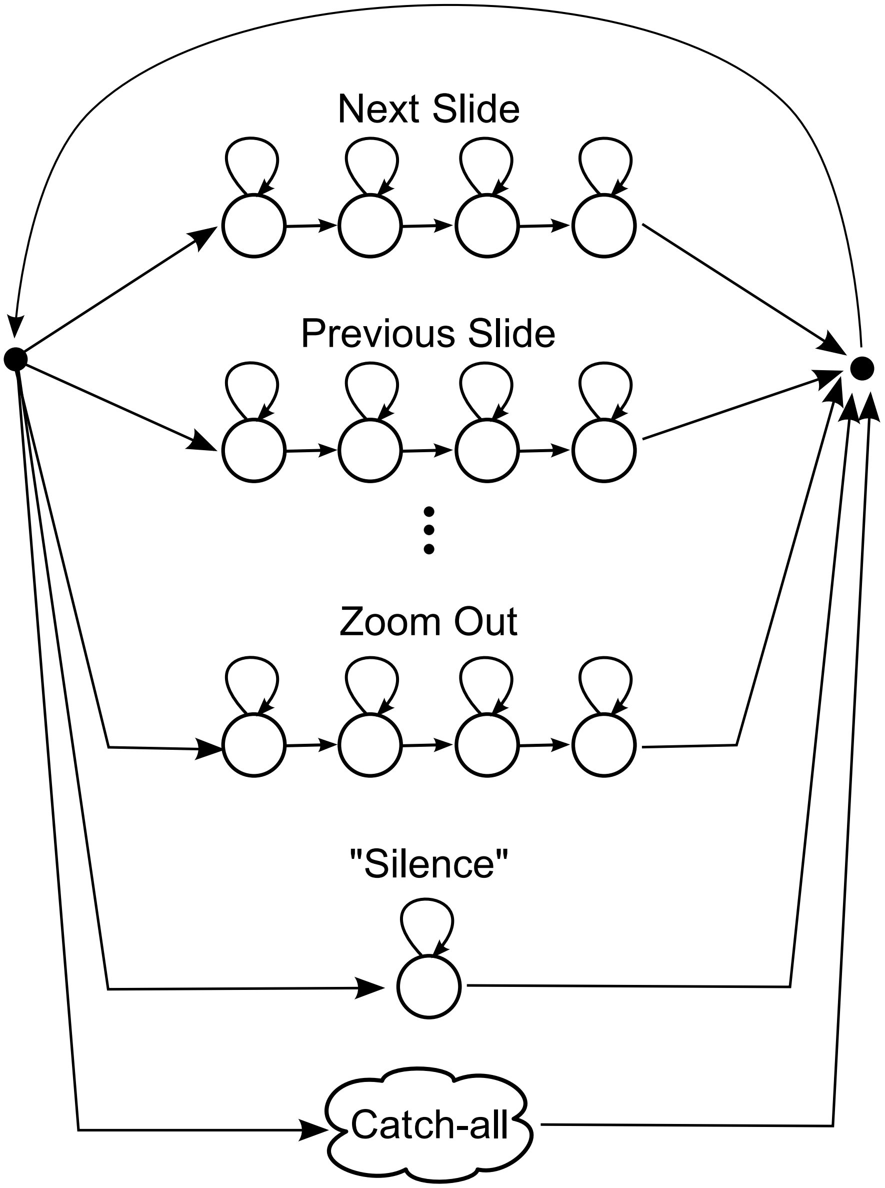


Figure 7.2: Maestro's gesture spotting network. Again, the black dots represent non-emitting states. The background process is factored into a "silence" and ergodic catch-all model. These models are connected in parallel to the gesture models.

Chapter 8

Gesture Spotting Results

This chapter presents results from a standardized user study which directly compares Maestro's original ad-hoc recognizer to the HMM-based approach described in the previous two chapters. In addition to presenting quantitative results for both recognizers, this study shows that both recognizers have very similar recognition characteristics. As such, we conclude that the lessons learned when using the ad-hoc recognizer (including the results of the deployment study described in Chapter 5) are equally applicable to a system which uses our DHMM-based gesture recognizer. Going forward, we recommend the DHMM-based approach since it is more generalizable, it follows a more principled approach, and it is more representative of the state-of-the-art in this field.

8.1 Training the gesture models and initial positive results

Chapter 6 described the 4-state left-right hidden Markov models that are used to model each of Maestro's gestures. In order to train these DHMMs, a considerable amount of data was required. The training data included approximately 100 isolated examples of each of the following 11 gestures (as depicted in figures 4.1 and 4.2 of Chapter 4):

> next slide, previous slide, undo, scroll down, scroll up, open carousel, close carousel, zoom in, zoom out, expand bullet and collapse bullet

In the case of context-sensitive gestures, such as expand, collapse, and zoom-in, training was conducted by randomly relocating the targets after each gesture performance. This avoids learning a model that is specific to a single location or slide layout.

Each of the gesture training examples was performed by one individual (the author of this document), and gestures were isolated by hand.

	Next	Previous	Undo	Scroll Up	Scroll Down	Open Carousel	Zoom In	Expand	Collapse	Zoom Out	Close Carousel
Next	87	4									
Previous		100									
Undo	1		104								
Scroll Up		2		99							
Scroll Down				4	91						
Open Carousel						101					
Zoom In							105				
Expand								105			
Collapse									105		
Zoom Out										106	
Close Carousel											100

Table 8.1: Aggregate confusion matrix, $C = [c_{ij}]$, resulting from the five-fold cross validation of the DHMM-based isolated gesture recognizer. Entry c_{ij} indicates the frequency with which gesture G_i was recognized as gesture G_j.

Having acquired approximately 100 examples of each gesture, model parameters were learned using the Baum-Welch reestimation procedure described in Chapter 6. For each gesture, five different models were learned (by randomizing initial conditions), and the one with the highest likelihood of generating the training data was selected. An initial inspection of these models reveals that the reestimation procedure seems to have well-captured the essence of each gesture. Several of these DHMMs are depicted graphically in figures 8.1 - 8.3.

In addition to examining the newly learned gesture models, we performed an initial test to verify the potential of the DHMM-based approach. Since the training data represents isolated gestures, it can be used to evaluate the performance of the DHMMs in an isolated gesture recognition task (Section 6.3). Of course, one cannot both train and evaluate the models using a common data set – instead, a five-fold cross validation procedure was used. When using five-fold cross validation, the training data for each gesture was partitioned into 5 sets of approximately equal cardinality. This was followed by 5 separate experiments. In each experiment, four sets were used for training, and the fifth was retained for the evaluation. These experiments revealed that between 98% and 99% of the isolated gestures were recognized correctly across each of the five folds. An aggregate of the confusion matrices for isolated recognition is presented in table 8.1. These positive results suggest that the models are able to accurately discriminate Maestro's gestures from one another.

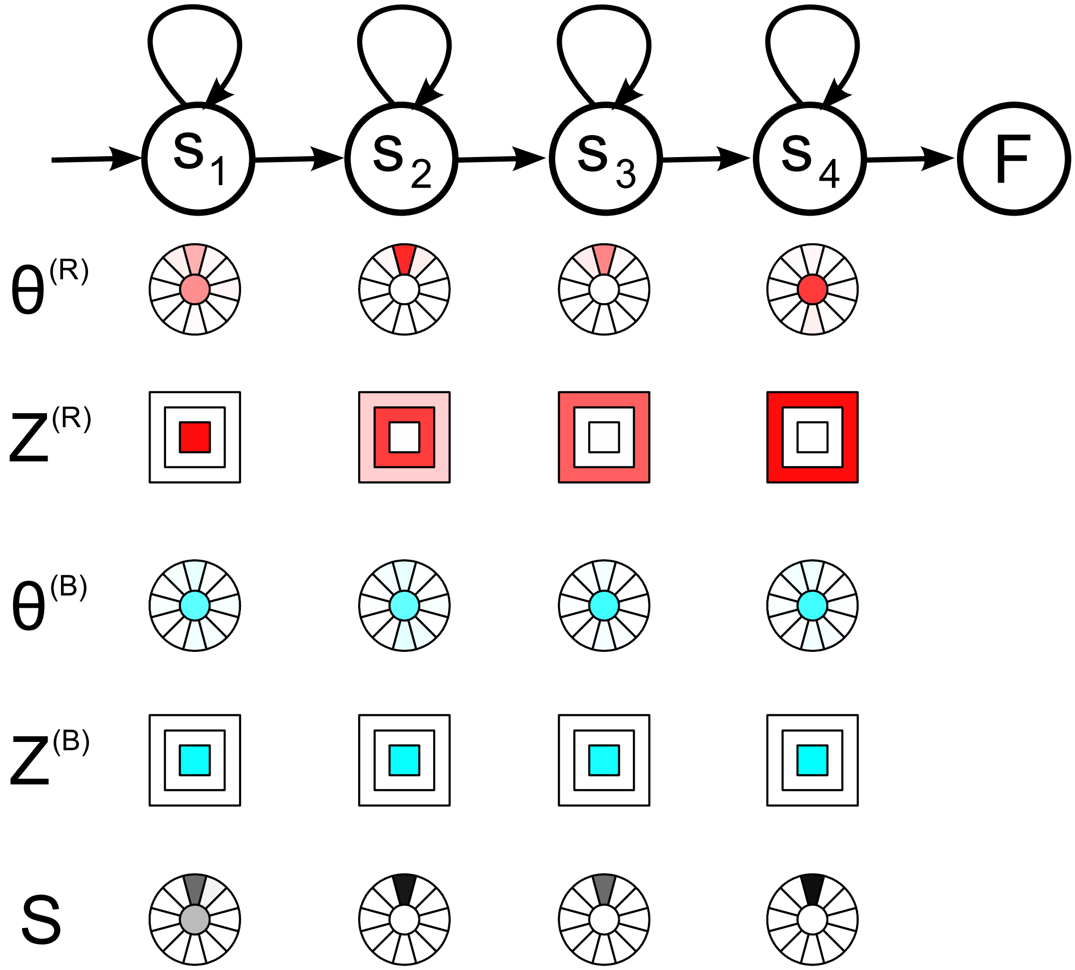


Figure 8.1: Visualization of the “scroll up” gesture’s DHMM. In this gesture, the red glove moves up while the blue glove remains stationary. Both gloves start together, in the center section of Maestro’s staging area.

In this figure, each cell represents a possible value for the discrete turning angles ($\Theta^{(R)}$ and $\Theta^{(B)}$), zones ($Z^{(R)}$ and $Z^{(B)}$), and spatial relation (S) features, as described in figures 6.3a and 6.3b of Chapter 6. The intensity of the shading of each cell represents the *marginal probability* of observing the value for the corresponding feature (e.g., $P[\Theta^{(R)} = x|Q = s_i]$, where x is the value and s_i is the state). For example, upward motion of the red glove has high marginal probability for the first 3 states; but, in the last state, the red glove is most likely at rest. Similarly, the spatial relation feature (which points from the blue glove towards the red glove) indicates that the red glove starts near the blue glove, and remains directly above the blue glove for the duration of the gesture.

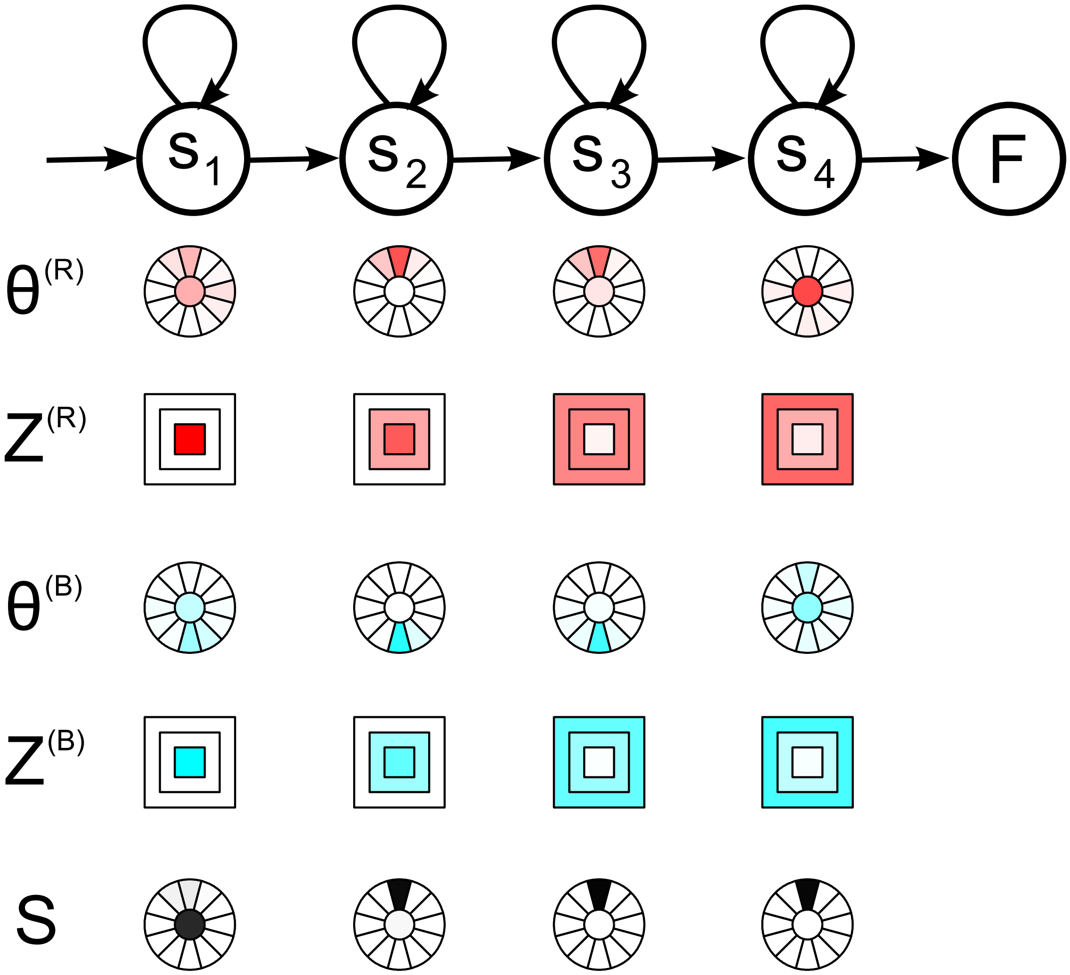


Figure 8.2: Visualization of the “zoom in” gesture’s DHMM. In this gesture, the red glove and blue glove move apart vertically. The hands start together, inside the bounds of the image being zoomed. See figure 8.1’s caption for a detailed explanation of this figure.

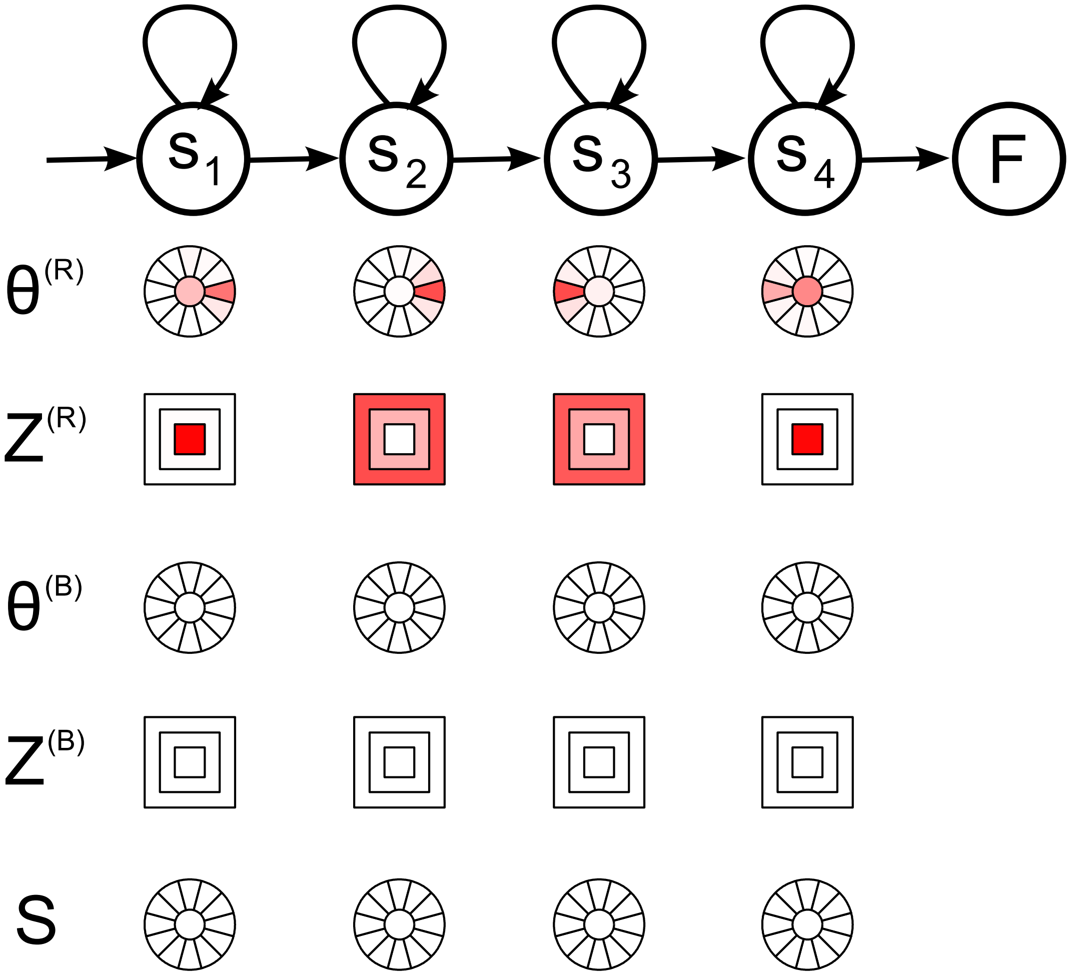


Figure 8.3: Visualization of the "undo" gesture's DHMM. In this gesture, the red glove moves first to the right and then to the left, starting in and returning to the lower section of Maestro's staging area. See figure 8.1's caption for a detailed explanation of this figure.

8.2 Results for gesture spotting

Motivated by the initial positive results of the isolated gesture recognition task, a set of formal experiments was conducted to compare the error rates of the DHMM-based gesture spotting approach to those of the ad-hoc recognizer. In order to obtain more general results, five individuals were recruited to participate in this study. Participants included 3 males and 2 females, aged 20-35 years old. Three of the individuals had previously participated in trials during the iterative design of Maestro, while two users had never used Maestro before. All five individuals are considered inexperienced users. Additionally, the experiment was repeated by an expert user (the author of this document).

In each of the six trials, users were asked to repeatedly perform each gesture until Maestro's original recognizer spotted at least 10 instances of the gesture. Each trial used a standardized set of presentation slides providing context against which the gestures were performed. While the standardized slides resembled a typical presentation, the slides were static and did not respond to gestures. Instead, an audible tone was used to provided real-time feedback to the participants; the tone sounded whenever the ad-hoc recognizer spotted a gesture.

In each experiment, the participant's hand trajectories were logged, and the trials were video recorded. Once the data was gathered, the logs could be replayed to simulate input. This, along with the static standardized slides, allowed a common data set to be tested against both the ad-hoc and HMM-based recognizers. The videos were then manually coded to establish a ground truth for the frequency with which each gesture was actually performed.

Confusion matrices for the participants are presented in tables 8.2a and 8.2b. Since gesture spotting must also be able to rule out non-gestures, the confusion matrix has an extra row and column compared to the confusion matrix for isolated gesture recognition (table 8.1). Cases where a gesture was confused for a non-gesture (the right-most column) correspond to false-negatives. Cases where non-gestures were mistaken for gestures (the bottom row) are false-positives.

For participants, the ad-hoc recognizer correctly spotted 86.4% of the gesture performances, and less than 1% of all detections were false-positives. This can be compared to the HMM-based approach, where the recognizer correctly spotted 84.6% of the gesture performances, and 3% of all detections were false positives. While the HMM-based approach scores slightly lower than the ad-hoc recognizer in both cases, the results differ only by a few percentage points. Both recognizers appear to error on the side of caution, having a strong bias toward false-negatives over false-positives. As noted in Chapter 5, this behavior is desirable in presentation systems since false-positives are more detrimental to the presentation. It should also be noted that these error rates compare favorably with those of similar systems. For example, Charade achieved an accuracy ranging from 72% to 84% for inexperienced users when using a modified Rubine recognizer [2].

The recognition results of the participants can be compared to the results of an

	Next	Previous	Undo	Scroll Up	Scroll Down	Open Carousel	Zoom In	Expand	Collapse	Zoom Out	Close Carousel	Non-gesture
Next	50											2
Previous	2	55										4
Undo			53									9
Scroll Up		6		45								6
Scroll Down	2				47							1
Open Carousel						48						12
Zoom In							49					5
Expand								51	1			16
Collapse									38			15
Zoom Out										50		4
Close Carousel											53	1
Non-gesture	1			3			1					–

(a) Confusion matrix for spotting participant gestures, when using the ad-hoc recognizer.

	Next	Previous	Undo	Scroll Up	Scroll Down	Open Carousel	Zoom In	Expand	Collapse	Zoom Out	Close Carousel	Non-gesture
Next	45											7
Previous		49										12
Undo			57									5
Scroll Up				39								18
Scroll Down					44							6
Open Carousel						60						
Zoom In							41					13
Expand								49	1			18
Collapse									38			15
Zoom Out										53		1
Close Carousel											54	
Non-gesture	1	2	4	1		1			7			–

(b) Confusion matrix for spotting participant gestures, when using the HMM-based recognizer.

Table 8.2: Confusion matrices for spotting participant gestures.

expert user. Results for the expert user are presented in tables 8.3a and 8.3b. For the expert user, the ad-hoc recognizer correctly spotted 96.4% of the gesture performances, and there were no false positives. Conversely, the HMM-based approach correctly spotted 95.7% of the gesture performances, with 4% of all detections resulting from false positives. Again, these results compare favorably with those of similar systems; CHARADE achieved an accuracy ranging from 90% to 98% for expert users [2], while FreeHandPresent [28] achieved an accuracy of 93% using a DHMM-based approach that is very similar to the approach used by Maestro (the main differences being Maestro's factored observation model, and Maestro's explicit modeling of "silence" – i.e., cases where neither hand is detected).

For the expert user, *all* false-positives occurred when performing the "collapse bullet" gesture. This same gesture also resulted in a high number of false-positives when the HMM-based approach was evaluated with non-expert participant data. In both cases, there were also a large number of false-negatives for this gesture. This suggests a problem with the "collapse bullet" gesture or its model, rather than a problem with the underlying gesture spotting approach.

8.3 Conclusion and future work

The ad-hoc and HMM-based recognizers are sufficiently similar so as to suggest that the lessons learned when using the ad-hoc recognizer (including the results of the deployment study described in Chapter 5) are equally applicable to a system using the HMM-based approach. Going forward, we recommend the HMM-based approach which is more generalizable, follows a more principled approach, and is arguably more representative of the state-of-the-art in this area. While the ad-hoc recognizer functions quite well in practice, it is not easily generalized to cope with new gestures or environments. In fact, adding new gestures (or improving the recognition of existing gestures) requires new heuristics to be developed on a case-by-case basis. When using an approach based on hidden Markov models, these same tasks can be achieved by simply collecting new training data and learning new models.

While the two gesture recognizers perform quite well, they have slightly different recognition characteristics. As noted earlier, false-positives are more likely to occur when using the HMM-based recognizer. It is possible that a better background model, or a different confidence measure, may reduce the occurrence of false-positives. More work must be done to explore these possibilities.

Additionally, more work must be done to determine if recognition results can be improved by using different model topologies for gestures. For example, some gestures may be better modeled using a 3-state HMM rather than a 4-state HMM. Alternatively, it is worth exploring model topologies in which one or more states can be skipped (i.e., increasing the Δ in the constrained jump topology, as described in Section 6.3). It is also worthwhile considering moving from an HMM with discrete observation vectors to one combining both discrete and continuous values.

	Next	Previous	Undo	Scroll Up	Scroll Down	Open Carousel	Zoom In	Expand	Collapse	Zoom Out	Close Carousel	Non-gesture
Next	10											
Previous		10										
Undo			10									
Scroll Up				10								
Scroll Down					10							
Open Carousel						10						
Zoom In							15					2
Expand								10				
Collapse									17			3
Zoom Out										15		
Close Carousel											16	
Non-gesture												–

(a) Confusion matrix for spotting expert gestures, when using the ad-hoc recognizer.

	Next	Previous	Undo	Scroll Up	Scroll Down	Open Carousel	Zoom In	Expand	Collapse	Zoom Out	Close Carousel	Non-gesture
Next	10											
Previous		10										
Undo			10									
Scroll Up				10								
Scroll Down					9							1
Open Carousel						10						
Zoom In							17					
Expand								10				
Collapse									15			5
Zoom Out										15		
Close Carousel											16	
Non-gesture									6			–

(b) Confusion matrix for spotting expert gestures, when using the HMM-based recognizer.

Table 8.3: Confusion matrices for spotting expert gestures.

Finally, we note that Maestro's "expert user", discussed above, is the same individual who generated most of the training data for learning the gesture models. It is likely that the high accuracy with which his gestures were spotted are the result of both experience, and the fact that various aspects of the models may be tailored to his unique motion characteristics. In the future, it will be important to separate these factors to determine if there is any advantage to personalizing the gesture models. This personalization has been suggested by others [61] and has proven to be an effective means to increasing accuracy in speech recognition applications.

Chapter 9

Conclusion

Gesture-based interaction has long been seen as a natural means of input for electronic presentation systems. However, gesture-based presentation systems have not been evaluated in real-world contexts. To address this issue, we designed and evaluated Maestro, a gesture-based presentation system which uses computer vision for gesture recognition. This work was presented in two parts. The first part served to motivate gesture-based presentation control, and to discuss the details of Maestro's design and evaluation. Importantly, the design was motivated by a small observational study of people giving talks, and the evaluation was conducted in a real-world setting over a two-week period.

Part II of this document presented a sophisticated gesture recognizer, which was based on discrete hidden Markov models. In comparison to Maestro's original ad-hoc recognizer, this new recognizer is more generalizable in the sense that new gestures can easily be added by training new models. Moreover, the HMM-based recognizer is more representative of the state-of-the-art in this field. Crucially, user trials have shown both recognizers have similar recognition characteristics. As such, we believe the conclusions presented in part I are equally applicable when using the more sophisticated recognizer described in part II. We now review both parts in turn, and then list some possibilities for future research.

9.1 Part I: Design and evaluation of Maestro

The first five chapters of this document presented the design and evaluation of Maestro, along with an overview of related work. In comparison to other gesture-based presentation systems in the literature, Maestro is distinguished by the following:

1. Maestro's design is directly influenced by an observational study examining the practices of presenters when giving talks. This study indicated that gestures are typically directed at slide content, and are primarily used to more effectively communicate the presentation material.

2. Maestro makes use of two classes of gestures, those that *navigate* the presentation (e.g., moving between slides), and those that operate on slide *content* (e.g., highlighting content when pointed to). Past research has not distinguished between these two classes of gestures, and has instead focused almost exclusively on using gestures for presentation navigation. Our findings suggest content-centric gestures are the most important in such a system.

3. Maestro was evaluated in a classroom setting, where it was used for a period of two weeks. To the best of our knowledge, this study constitutes the first real-world, long-term evaluation of such a system.

Additionally, the deployment study suggests that gesture-based control can noticeably alter the dynamics of a presentation in ways that are not always desirable. In particular, sensing needs can reduce the mobility of the presenter leading to the "anchoring problem". Additionally, since gestures are performed in close proximity to the screen, the projected content may not fit within the presenter's field of view. This can lead to awkward breaks in the presentation where the presenter must step away from the screen to view each slide in its entirety. Finally, the interface can introduce "no-fly zones": regions in which the presenter may not enter without the risk of accidentally issuing a command, or otherwise distracting the audience. These findings were unexpected, have not been previously reported in the literature, and help set an agenda for future research in this area.

9.2 Part II: Gesture recognition with discrete hidden Markov models

In support of rapid prototyping, Maestro's original gesture recognizer was heuristic in nature, and relied on manually tuned gesture templates. Part II of this document presented a more sophisticated gesture recognizer based on discrete hidden Markov models. Importantly, Maestro's DHMMs make use of a factored observation model that allows modeling of both one and two-handed gestures, and which directly models missing observations. The factored observation model greatly reduces the number of parameters that would otherwise need to be learned to model Maestro's gestures.

Upon training the DHMM-based recognizer, both this recognizer and Maestro's original ad-hoc recognizer were evaluated in a controlled laboratory setting. In this experiment, five participants and one expert user each performed at least ten examples of every gesture. Both the ad-hoc and DHMM-based recognizers scored favorably in comparison to similar systems in the literature. Importantly, both recognizers also exhibited very similar recognition characteristics. The ad-hoc recognizer accurately spotted 86% of gestures for new users, increasing to 96% for the expert user; while the DHMM-based recognizer accurately spotted 85% of gestures for new users, again increasing to 96% for the expert. This suggests that the conclusions drawn from Maestro's deployment study continue to be applicable when using more sophisticated gesture recognizers.

9.3 Future work

While the development of the DHMM-based gesture recognizer already represents an evolution of Maestro's original design, there are many opportunities for continued research in this project. As with many other aspects of this document, these research possibilities relate to human-computer interaction, computer vision, as well as gesture recognition. We discuss these possibilities next.

First and foremost, our work on Maestro strongly suggests exploring multimodal interactions with presentations. Specifically, we recommend that rich interactions be achieved via gestures, and efficient navigation of the presentation be attained through input devices such as remote controls.

Our work also suggests exploring new forms of interaction that may enhance one's ability to emphasize or better communicate the content within projected slides. As was noted earlier, in Part I, one possibility is to explore gesture-based interactions with mathematical plots and simulations. Some of these possibilities have already been partially explored by Douglas Zongker and others [70], but past research has not explored the use of gesture-based interaction in this context.

There also exists the possibility of automating Maestro's initial calibration procedure, which is required prior to interacting with the projected slideshow. While Maestro's current calibration process simply involves specifying the 4 corners of the display, it nonetheless is an additional step that is not required by PowerPoint or similar presentation systems. Work by Sukthankar *et al.* [55] demonstrates how calibration can be achieved automatically by projecting calibration patterns onto the screen, and using corner detection techniques from the computer vision community to recover the necessary calibration parameters.

Along these lines, it may also be possible to develop hand tracking techniques that do not rely on the colored gloves. Work by Maria Hilario *et al.* demonstrates how knowledge of the projected background, and a model for the camera's color response, can allow detection of objects that occlude the display [14]. While this approach provides only the presenter's contour, this extra information may allow the hands to be detected and tracked more easily.

Finally, we would also like to develop more accurate background models that better account for the presenter's hand motion when he or she is not performing a gesture. An improved background model may improve the recognition results of the DHMM-based approach, and is a compelling possibility for more theoretical research in this area.

References

[1] Anonymous. Field manual: Visual signals. Technical Report FM 21-60, United States Armed Forces, September 1987. 17

[2] Thomas Baudel and Michel Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993. 1, 2, 7, 10, 13, 27, 33, 45, 84, 86

[3] M. J. Black and A. D. Jepson. Recognizing temporal trajectories using the condensation algorithm. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 16, Washington, DC, USA, 1998. IEEE Computer Society. 13, 59

[4] G. Bouwman, L. Boves, and J. Koolwaaij. Weighting phone confidence measures for automatic speech recognition. In *Proceedings of the COST249 Workshop on Voice Operated Telecom Services*, pages 59–62, 2000. 73, 74

[5] Xiang Cao, Eyal Ofek, and David Vronay. Evaluation of alternative presentation control techniques. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1248–1251, New York, NY, USA, 2005. ACM. 1, 3, 9, 20, 33, 40, 55

[6] Kelvin Cheng and Kevin Pulo. Direct interaction with large-scale display systems using infrared laser tracking devices. In *APVis '03: Proceedings of the Asia-Pacific symposium on Information visualisation*, pages 67–74, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. 9

[7] P. Chiu, Q. Liu, J. Boreczky, J. Foote, T. Fuse, D. Kimber, S. Lertsihichai, and C. Liao. Manipulating and annotating slides in a multi-display environment. In *Proceedings of INTERACT 2003*, 2003. 2

[8] Arie Croitoru, Peggy Agouris, and Anthony Stefanidis. 3d trajectory matching by pose normalization. In *GIS '05: Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 153–162, New York, NY, USA, 2005. ACM. 65

[9] CyberNet. Gesture storm. `http://www.gesturestorm.com/`, September 2008. 12

[10] Trevor Darrell and Alex Pentland. Space - time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 335–340, June 1993. 59

[11] S. Eickeler, A. Kosmala, and G. Rigoll. Hidden markov model based continuous online gesture recognition. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1206–1208 vol.2, 1998. 60, 74, 75

[12] A. Fourney and R. Mann. Non-accidental features for gesture spotting. In *Computer and Robot Vision, 2009. CRV '09. Sixth Canadian Conference on*, June 2009. 28

[13] Trevor Harley. *The Psychology of Language*, chapter 1, page 4. Psychology Press, 2 edition, 2001. 15

[14] Maria Nadia Hilario and Jeremy R Cooperstock. Occlusion detection for front-projected interactive displays. In *Second International Conference on Pervasive Computing*, Linz/Vienna, Austria, 2004. Springer Berlin Heidelberg. 34, 91

[15] iMatte. iMatte - Technologies. `http://www.imatte.com/index.html`, September 2008. 22, 65

[16] T. Isobe and J. Takahashi. A new cohort normalization using local acoustic information for speaker verification. In *ICASSP '99: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference*, pages 841–844, Washington, DC, USA, 1999. IEEE Computer Society. 74

[17] Robert J. K. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans. Inf. Syst.*, 9(2):152–169, 1991. 11, 12

[18] A. Jepson and W. Richards. What makes a good feature? In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 89–126. Cambridge University Press, 1995. Also MIT AI Memo 1356 (1992). 28

[19] Hui Jiang. Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4):455–470, 2005. 72, 73

[20] S.X. Ju, M.J. Black, S. Minneman, and D. Kimber. Analysis of gesture and action in technical talks for video indexing. pages 595–601, Jun 1997. 14

[21] S.O. Kamppari and T.J. Hazen. Word and phone level acoustic confidence scoring. In *ICASSP IEEE INT CONF ACOUST SPEECH SIGNAL PROCESS PROC*, volume 3, pages 1799–1802, 2000. 73

[22] Hyun Kang, Chang Woo Lee, and Keechul Jung. Recognition-based gesture spotting in video games. *Pattern Recogn. Lett.*, 25(15):1701–1714, 2004. 74

[23] Maria Karam. *A framework for research and design of gesture-based human computer interactions*. PhD thesis, University of Southampton, November 2006. 15

[24] Daehwan Kim, Jinyoung Song, and Daijin Kim. Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms. *Pattern Recogn.*, 40(11):3012–3026, 2007. 13

[25] Carsten Kirstein and Heinrich Mueller. Interaction with a projection screen using a camera-tracked laser pointer. In *MMM '98: Proceedings of the 1998 Conference on MultiMedia Modeling*, page 191, Washington, DC, USA, 1998. IEEE Computer Society. 9

[26] Joel Lanir, Kellogg S. Booth, and Anthony Tang. Multipresenter: a presentation system for (very) large display surfaces. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 519–528, New York, NY, USA, 2008. ACM. 1

[27] Paul J. Lavrakas, editor. *Encyclopedia of Survey Research Methods*, volume 2, page 429. Sage Publications, 2008. 38, 39

[28] Hyeon-Kyu Lee and Jin H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(10):961–973, 1999. 2, 3, 13, 45, 60, 64, 65, 66, 72, 75, 76, 86

[29] Philippa Levy. Interactive whiteboards in learning and teaching in two sheffield schools: a developmental study. `http://www.imatte.com/index.html`, January 2002. 10

[30] A. Licsar and T. Sziranyi. Hand gesture recognition in camera-projector system. In *CVHCI04*, pages 83–93, 2004. 11, 34

[31] R. Mann and A. Jepson. Non-accidental features in learning. In *Proceedings of AAAI Fall Symposium on Machine Learning in Vision*, October 1993. 28

[32] Tobias P. Mann. Numerically stable hidden markov model. `http://bozeman.genome.washington.edu/compbio/mbt599_2006/hmm_scaling_revised.pdf`, February 2006. 61

[33] David McNeill. *Hand and Mind: What gestures reveal about thought*, chapter 3, page 92. University of Chicago Press, 1992. 18

[34] David McNeill. *Gesture and Thought*, chapter 2, pages 39–40. The University of Chicago Press, 2005. 15, 17

[35] McNeill Lab. Mcneill lab center for gesture and speech research. `http://mcneilllab.uchicago.edu`, June 2009. 15

[36] Microsoft. Microsoft delivers office live workspace beta. `http://www.microsoft.com/presspass/features/2007/dec07/12-10officeLiveWorkspace.mspx`, December 2007. 1

[37] Microsoft. Microsoft office powerpoint. `http://office.microsoft.com/powerpoint/`, July 2009. 1

[38] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(3):311–324, 2007. 60

[39] Gemma Moss, Carey Jewitt, Ros Levaaic, Vicky Armstrong, Alejandra Cardini, and Frances Castle. The interactive whiteboards, pedagogy and pupil performance evaluation: An evaluation of the schools whiteboard expansion (swe) project: London challenge. Technical Report Research Report RR816, Institute of Education, University of London, 2007. 3

[40] Les Nelson, Satoshi Ichimura, Elin Ronby Pedersen, and Lia Adams. Palette: a paper interface for giving presentations. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 354–361, New York, NY, USA, 1999. ACM. 1

[41] D. Okumura, S. Uchida, and H. Sakoe. An hmm implementation for on-line handwriting recognition based on pen-coordinate feature and pen-direction feature. *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 26–30 Vol. 1, Aug.-1 Sept. 2005. 65

[42] Dan R. Olsen and Travis Nielsen. Laser pointer interaction. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–22, New York, NY, USA, 2001. ACM Press. 9

[43] Ian Parker. Absolute powerpoint: Can a software package edit our thoughts?, May 28 2001. 1

[44] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997. 15, 60

[45] A.M. Peinado, J.M. Lopez, V.E. Sanchez, J.C. Segura, and A.J. Rubio Ayuso. Improvements in hmm-based isolated word recognition system. *Communications, Speech and Vision, IEE Proceedings I*, 138(3):201–206, June 1991. 67, 72, 73

[46] Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCullough, and Rashid Ansari. Multimodal human

discourse: gesture and speech. *ACM Trans. Comput.-Hum. Interact.*, 9(3):171–193, 2002. 15, 16

[47] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990. 60, 61, 64, 67

[48] Gerhard Rigoll, Andreas Kosmala, and Stefan Eickeler. High performance real-time gesture recognition using hidden markov models. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 69–80, London, UK, 1998. Springer-Verlag. 60, 64

[49] R.C. Rose and D.B. Paul. A hidden markov model based keyword recognition system. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, pages 129–132 vol.1, Apr 1990. 72

[50] A.E. Rosenberg, J. DeLong, C.H. Lee, B.H. Juang, and F.K. Soong. The use of cohort normalized scores for speaker verification. In *Second International Conference on Spoken Language Processing*. ISCA, 1992. 74

[51] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. Shadow reaching: a new perspective on interaction for large displays. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 53–56, New York, NY, USA, 2007. ACM. 22

[52] Heather J. Smith, Steve Higgins, Kate Wall, and Jen Miller. Interactive whiteboards: boon or bandwagon? a critical review of the literature. *Journal of Computer Assisted Learning*, 21(2):91–101, 2005. 10

[53] Noi Sukaviriya, Rick Kjeldsen, Claudio Pinhanez, Lijun Tang, Anthony Levas, Gopal Pingali, and Mark Podlaseck. A portable system for anywhere interactions. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 789–790, New York, NY, USA, 2004. ACM. 11

[54] Rahul Sukthankar, Robert Stockton, and Matthew Mullin. Self-calibrating camera-assisted presentation interface. In *Proceedings of International Conference on Control, Automation, Robotics and Computer Vision*, December 2000. 2, 9, 71

[55] Rahul Sukthankar, Robert G. Stockton, and Matthew D. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. *iccv*, 01:247, 2001. 1, 9, 11, 13, 34, 91

[56] Desney S. Tan and Randy Pausch. Pre-emptive shadows: eliminating the blinding light from projectors. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 682–683, New York, NY, USA, 2002. ACM. 22

[57] Richard A. Tennant and Marianne Gluszak Brown. *The American Sign Language handshape dictionary.* Gallaudet University Press, 1998. 17

[58] Edward R. Tufte. *The Cognitive Style of PowerPoint: Pitching Out Corrupts Within, Second Edition.* Graphis Pr, 2006. 29

[59] Christian von Hardenberg and François Bérard. Bare-hand human-computer interaction. In *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, New York, NY, USA, 2001. ACM. 2, 11, 24

[60] Eric W. Weisstein. Inverse tangent. From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/InverseTangent.html`, July 2009. 66

[61] Alan Wexelblat. Research challenges in gesture: Open issues and unsolved problems. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 1–11, London, UK, 1998. Springer-Verlag. 15, 88

[62] L.D. Wilcox and M.A. Bush. Training and search algorithms for an interactive wordspotting system. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 2, pages 97–100 vol.2, Mar 1992. 75

[63] J.G Wilpon, C.H. Lee, and L.R. Rabiner. Application of hidden markov models for recognition of a limitedset of words in unconstrained speech. In *International Conference on Acoustics, Speech, and Signal Processing. ICASSP-89*, 1989. 72, 73, 74

[64] J.G. Wilpon, L.R. Rabiner, C.-H. Lee, and E.R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(11):1870–1878, Nov 1990. 74, 76

[65] Andrew D. Wilson and Aaron F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):884–900, 1999. 60

[66] Jie Yang and Yangsheng Xu. Hidden markov model for gesture recognition. Technical Report CMU-RI-TR-94-10, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1994. 60, 64, 67, 75

[67] Ming Hsuan Yang, Narendra Ahuja, and Mark Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1061–1074, 2002. 59

[68] Ho-Sub Yoon, Jung Soh, Younglae J. Bae, and Hyun Seung Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491–1501, 2001. 11

[69] Sheryl R. Young. Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words. Technical Report CMU-CS-94-157, Pittsburgh, PA, USA, 1994. 72, 76

[70] Douglas E. Zongker and David H. Salesin. On creating animated presentations. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 298–308, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 1, 57, 91