The Presentation Maestro: Direct Manipulation Through Gesture Alone

Adam Fourney
afourney@cs.uwaterloo.ca

Michael Terry
mterry@cs.uwaterloo.ca

Richard Mann
mannr@uwaterloo.ca

David R. Cheriton School of Computer Science
University of Waterloo

Technical Report CS-2009-14

ABSTRACT

Past research suggests a number of benefits to using hand-based interaction when interacting with electronic presentations. This paper introduces *Maestro*, a computer-vision based presentation system that uses hand gestures to allow fine-grained interaction with the contents of a projected slideshow. Maestro employs a single web camera and no other physical mediators. The contributions of this paper lie in robustly solving the gesture segmentation problem inherent in using only computer vision as input, and in a set of feedback mechanisms designed to scaffold use of the recognition system without interfering with the visual presentation of content. Specifically, Maestro employs *bimanual* cues, *spatial and content-based context*, *hand roles*, and, in some case, *dwell time* to segment gestures. These strong segmentation cues are robust with respect to false positives and allow the actual gestures themselves to more directly match the action to be performed. They also result in a direct manipulation-style interface built on gestures alone. To assist with use, Maestro introduces a *feedback comet*, an augmented cursor that provides mnemonics for gestures and feedback to help govern gesture speeds to those conducive to gesture recognition. We present the design of the system and lessons learned from user evaluations.

1 INTRODUCTION

An estimated 30 million presentations are given each day, making presentation software a critical application for millions of users [14]. Numerous research efforts have explored ways of improving this common activity. For example, Palette [12] enables presenters to randomly access slides using tangible cards, while Time Aura [11] provides ambient cues to help presenters pace themselves. In this paper, we are most concerned with methods of *interacting* with a presentation system and how this interaction can be provided in a way that serves to enhance, rather than detract from, the actual presentation.

Previous research by Cao et al. [2] has examined how interaction mechanisms can affect the act of giving a presentation. Using a Wizard-of-Oz study, they compared the effects of three alternative methods for interacting with a presentation system: Bare-handed interaction combined with a touch-sensitive surface; a laser pointer augmented with a button; and the use of a regular keyboard and mouse. The study found that both presenters and audience members prefer the use of bare-hand gestures, with audience members indicating that bare-hand interaction results in presenters using "a more personalized, humanized, story-telling style." These results provide strong motivation to explore bare-handed interaction techniques to control presentation systems.

A number of systems demonstrate the possibility of providing hand-based input to presentation systems using devices such as data gloves [1], touch-sensitive surfaces [2, 15], and computer vision (CV) [16]. Of the approaches, computer vision-based techniques are arguably the least expensive, especially given the ubiquity of inexpensive webcams. Motivated by the existence of this low-cost input technology, we focus exclusively on the challenges of enabling bare-handed interaction using only computer vision and no additional input.

The unique challenges in providing bare-handed interaction in a presentation context have been previously articulated by Baudel and Beaudouin-Lafon in their work developing Charade [1]. Specifically, any such system must be able to reliably segment and recognize gestures from a constant stream of input, while providing sufficient feedback to guide system use. The Charade system solved the segmentation and recognition problem using a data glove and gestures differentiated by hand postures and finger orientations. Work by Hardenberg et al. [16] demonstrates that some of these same features can be detected using computer vision techniques. However, in both cases, the gestures developed did not lend themselves to fine-grained, *direct* interaction with the projected content. Instead, they were largely *context independent* gestures that operated on the entire slide, rather than elements of the slide. As a result, users are not provided with the same degree of expressiveness possible with a pointing device. Additionally, these systems provided limited feedback, though such feedback is essential for any recognition-based interface. These open issues indicate the need for further research examining how to support fine-grained, direct interaction using gestures alone. They also suggest the need for feedback mechanisms that guide system use, without significantly altering the presentation itself.

This paper presents *Maestro*, a computer vision-based presentation system that allows rich, fine-grained *direct* interaction with projected content. For input, the system requires only the addition of a single webcam. Using gestures alone, users are able to interact with individual elements of slides, creating a direct manipulation interface that does not require the use of physical mediators. The contributions of this work lie in robustly solving the gesture recognition problem in the context of giving a presentation, and in a set of feedback mechanisms that assist in learning the system, scaffolding gestures as they are performed, and echoing actions after they are executed. We describe each contribution in turn.

The primary challenge in using only computer vision as input is to create a set of gestures that can be reliably segmented and recognized, without the introduction of physical mediators or an artificial stroke alphabet dissimilar to the actions to be performed. Maestro solves these problems by introducing a set of unique cues to signal the start and end of a gesture, where these cues are closely tied to the objects of interest. In particular, gestures in Maestro start and end using a combination of *bimanual* cues, *spatial and content-based context*, *hand roles*, and, in some cases, *dwell time*. For example, to expand a bullet point to show its (hidden) children, a user places both hands next to the bullet point, then moves one hand down. In this scenario, the bullet point supplies *context* and the *proximity* of the hands indicates that it is the object of interest. The result is a direct manipulation-style of interaction using gestures alone.

Importantly, Maestro's use of start and end cues allows the actual gestures to more naturally coincide with the activity. That is, the actual paths of each gesture do not need to be uniquely distinct

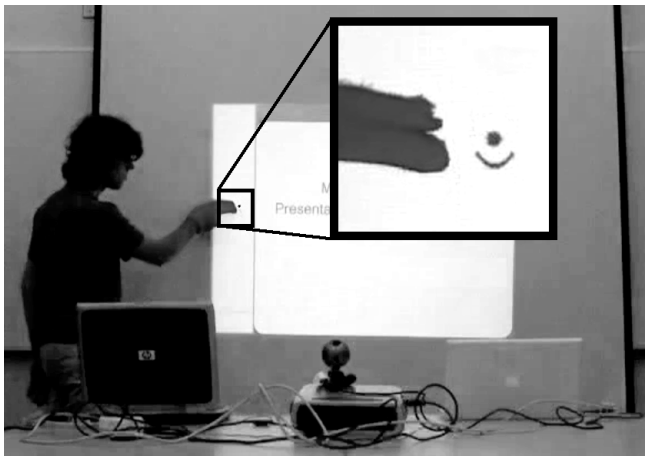

Figure 1: A close-up view of Maestro's *feedback comet*. The inset image shows the comet's head, a small dot indicating where the system thinks the presenter is pointing. Directly below the head, a gesture mnemonic is visible. In this case, the mnemonic is a convex curve indicating that the next slide can be accessed by moving the hand downward.

with respect to one another, as is the case with stroke languages where commands are differentiated solely on paths (e.g., Grafitti [13]). As a result, they can more naturally align with the action to be performed. For example, expanding a bullet point is performed with a downward motion, which results in bullet points appearing below the parent bullet point. Segmenting based on these cues also has the desirable result that the actual gesture paths can be *reused*, as long as they employ different start or end cues. For example, the actual gestures for scrolling a slide, navigating to the next slide, or expanding a bullet point are all identical; they differ only by the spatial and contextual cues demarcating the start of the gesture.

Maestro also makes contributions in the set of feedback mechanisms it offers. Maestro provides three types of feedback: pre-gesture mnemonics to indicate what commands are possible, and where; in-gesture feedback to help the user govern their gesture speed to a speed conducive to recognition; and post-gesture feedback to indicate when a command has been invoked. The first two forms of feedback are provided through a *feedback comet*, a small, context-sensitive cursor that follows the hands (Figure 1). Importantly, all of these cues are designed to minimally alter the projected content.

The remainder of the paper is structured as follows: we begin by presenting related-work, followed by a brief description of an observational study which we conducted. Following the observational study, we present key elements of Maestro's design, and we then discuss lessons learned from a small-scale user evaluation. Future research possibilities are then presented in the final section of this paper.

2 Background

Presentation systems enhance one's ability to effectively communicate information. In this paper, we are concerned with *interaction mechanisms* of such systems, and how such mechanisms can be designed to seamlessly integrate with the way people naturally give presentations. With this focus in mind, we first review work related to interaction mechanisms for presentation systems.

As previously mentioned, Cao *et al.* studied the impact input modality has on the effectiveness of giving a presentation [2]. In this Wizard-of-Oz style study, 6 individuals were asked to present 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 with a button, or hand gestures and a touch-sensitive surface. The audience members were asked to rate each presentation for clearness, efficiency and attractiveness using a numeric 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. The results of this study strongly argue for the benefits of gesture-based interfaces to presentation systems. Numerous efforts have explored this style of interaction in this context and can be classified as either *user-segmented systems* or *continuous motion systems*.

2.1 User-Segmented Systems

User-segmented systems require presenters to signal the start and/or the end of a gesture by depressing a button on a remote or similar device. Two commercial systems fall into this category. iSkia, distributed by the iMatte company, uses hand tracking and a wireless remote to emulate a mouse [7]. While not a gesture-based interface per se (since the hand emulates a mouse), it nonetheless provides more direct interaction than using a mouse. GestureStorm allows television network meteorologists to interact with their weather maps using gestures [4]. For example, circling an area of the map might cause the system to zoom into a particular region or district. Touch-sensitive surfaces, such as SmartBoards, also enable user-segmented gesture input [15].

2.2 Continuous Motion Systems

Continuous motion systems automatically *segment* gestures from a constant stream of input, an activity which significantly increases the complexity of the gesture recognition system. A number of continuous motion presentation systems have been described in the literature.

CHARADE is a continuous motion system that tracks hands and hand postures via a 3D tracking system and a DataGlove [1]. In this system, users articulate gestures using hand and finger orientation, and movement across an "active space" – a region of the presentation within which the system looks for gestures. This system also provides limited feedback to users and the ability to "undo" accidental commands, a facility the implementers found essential to accommodate recognition errors.

A number of continuous motion systems employ computer vision as the basis of their gesture input. A system by Von Hardenberg and Bernard demonstrates the use of hand-poses as input [16]. For example, extending two fingers instructs the presentation to advance to the next slide, while three outstretched fingers signals the system to return to the previous slide. Another system, PowerGesture [10], provides a gesture-based front-end to Microsoft PowerPoint. Ten separate gestures can be recognized in streams of continuous hand motion. These gestures can be thought of as stroke-based gestures, where each gesture is defined by a unique path in space. With this system, users can manipulate the presentation (e.g., navigating back and forth in the slide, or quitting the presentation), but cannot interact with individual elements.

Given the ubiquity of inexpensive webcams, computer vision-based gesture interfaces are particularly compeling for providing input. Previous work in continuous motion systems demonstrate the feasibility of this approach, but a number of open issues remain.

2.3 Open Issues

Previous continuous motion systems have typically employed the use of a relatively artificial gesture language, where segmentation of gestures is supported either using arbitrary cues (e.g., extending two or three fingers) or via unique paths in space. There are two primary issues with these approaches. First, they require learning a set of gestures that lack a natural mapping to the actual actions

performed. For example, extending two fingers has little correspondence to the notion of moving to the next slide. Charade's approach of using finger and hand orientation also suffers from this problem. Second, these approaches do not cleanly support a direct manipulation-style of interaction, preventing one from directly interacting with individual elements of projected content.

Past research has also largely ignored the issue of feedback to end-users. However, feedback is critical to these recognition-based interfaces. Three needs arise in this context: Users should know what commands are available and when they can be invoked with gestures; users should receive feedback as they perform a gesture; and they should know when an action has occurred so they can recover should recognition errors arise. While simply stated, providing feedback in this context is made more difficult by the fact that any such feedback should not interfere with, or significantly affect, the presentation itself.

Given the positive benefit observed for bare-handed interaction, along with the open research problems associated with a gesture-based interface using only computer vision as input, we set out to create a gesture-based presentation system that: leverages existing, natural gestures when giving a presentation; allows for fine-grained, direct interaction with individual elements of the presentation; and provides appropriate feedback to scaffold its use. We turn now to an observational study conducted to inform the design of this system.

3 Observational Study

To inform our work, we began by observing videos of tech talks posted on Google's "Tech Talks at Google" website [5]. These talks showcased a wide range of presentation styles, but also revealed common behaviors.

In the videos, we identified a tendency for presenters to interact with the projected content, even though the content was, in effect, static (i.e., it didn't respond to physical interactions with it). We observed speakers interact with the content for the following communicative purposes:

- To point at material currently being discussed (e.g., to highlight it). In this case, presenters hovered *next* to a point
- To "crop" or frame content to single it out from other content. Framing content involved the use of one or two hands
- To show relationships by tracing lines (e.g., following arrows in a flowchart or tracing lines in a line graph)
- To highlight keywords in definitions, theorems, or other text by physically underline the words

The results from this study echo those of Cao *et al.* [2], where there appears to be a natural tendency to directly interact with content to enhance one's presentation. They also suggest the value in allowing direct interaction with the presentation system via the projected content, as speakers naturally orient themselves next to the projection screen.

We also observed that presenters use hands interchangeably and do not rely on hand posture when performing the actions listed above. This suggests that any gesture-based system should avoid the use of artificially-imposed hand postures (e.g., requiring one to orient the hand up to signal the start of a gesture) since presenters do not naturally use hand orientation when giving presentations.

These natural gestures and observations inspired us in the design of our own gesture language. We turn now to a description of our system.

4 Design

Maestro is a computer vision-based presentation system controlled entirely by hand gestures. The system's only method of input is a webcam; no additional physical input devices are necessary. This particular configuration raises a number of design challenges unique to this particular context (i.e., giving a presentation). We first describe the goals of the system and how these give rise to particular challenges, then describe Maestro in detail, indicating how it addresses the goals we set forth.

4.1 Design Desiderata

In the design of Maestro, we sought to meet the following ideals:

1. The system should not influence desired visual presentation of material, including choice of content, its visual appearance, or its layout. Furthermore, the system should not introduce artifacts into the presentation purely for the purpose of supporting interaction
2. The system should not require any physical input system other than a webcam
3. The system should support coarse- and fine-grained interaction with slides and their content using gestures that map closely onto common, existing, "natural" gestures
4. Users should not be restricted in how they deliver the presentation; they should be able to freely move around space and gesture naturally to the audience
5. The system should be easily learned, provide affordances for its use, provide appropriate feedback during use, and support recovery from recognition errors

In examining the list, it should be clear that optimizing any one goal typically has the effect of impacting other goals. For example, goal #1 (only project presentation content) directly interferes with goal #5 (provide affordances). Likewise, goals 2 (no physical hardware beyond a webcam) and 3 (employ natural gestures) directly conflict, since identifying gestures without explicit segmentation is difficult. These goals have not been explicitly delineated in past work, and thus serve as a set of ideals to strive for in such systems. We now describe Maestro in detail, noting how we addressed the desiderata above.

4.2 Basic System Setup

Maestro makes use of a regular webcam, two laptop computers, and a pair of color-contrasting gloves. One laptop serves to process video, while the other runs the custom-built presentation software. A second laptop is necessary only because the computer vision algorithms are currently run in an interpreted environment (Matlab); a more efficient optimization would require only one computer, and is well within the realm of possibility for any modern laptop.

Maestro detects and tracks hands via two brightly colored gloves, one red, one blue. Detection is achieved using simple color thresholding techniques, while tracking is accomplished through the continuous detection of the gloves from frame to frame. Because hand tracking is currently based on the colors of the gloves, Maestro is limited to displaying monochromatic slides. As more robust hand-tracking algorithms are developed in the computer vision community, the system should be able to operate without colored gloves, removing this restriction. Importantly, Maestro's gesture recognition system allows users to perform gestures with either hand. This ensures that the gesture language is compatible with hand tracking algorithms where the hands are not as easily differentiated.

Before a user can begin issuing commands, Maestro's visual system is calibrated using a two-step process. First, a homography is established between the image registered by the camera and the image projected by the LCD display. This allows hand positions to be expressed in the coordinate system used by the presentation

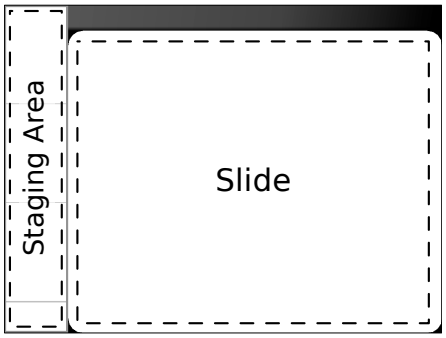

Figure 2: Maestro's layout and staging area.

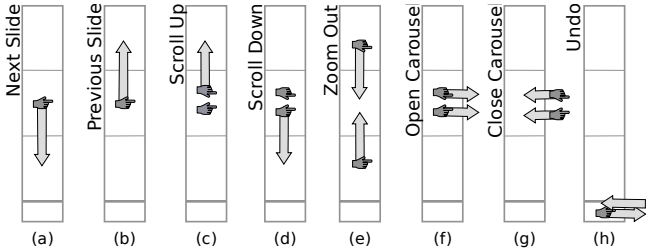

Figure 3: Gestures performed within Maestro's staging area.

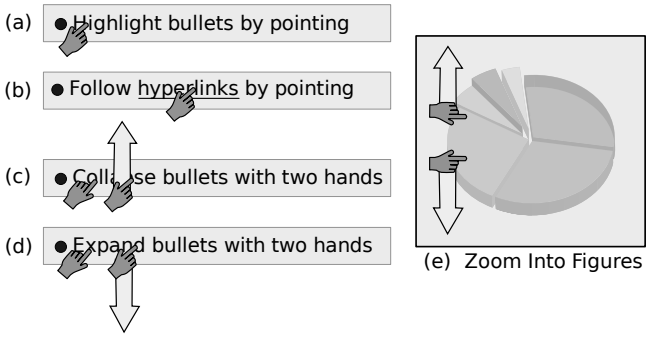

Figure 4: Gestures performed *directly* on slide content.

software. The homography is established by having the user manually locate the 4 corners of the display as seen in the first frame returned by the camera. Second, the user calibrates the camera's white balance by identifying a region of the display that is white. The slide background usually serves this purpose. Establishing a proper white balance increases the likelihood that the default glove color thresholds will be sufficient for detecting the hands.

Projected content is augmented with a small staging area located to the side (shown to the left in figure 2). The staging area allows users to interact with the overall presentation, as opposed to individual elements in the slide. Three horizontal divider lines define four regions in the staging area, an upper, middle, and lower region, which collectively allow the presenter to navigate content (e.g., scrolling slides, navigating to the next/previous slide, and viewing a slide carousel to select slides). A small bottom-most region, the "basement" region, overlaps the staging area and bottom of the slide region. This fourth region is used in conjunction with the "undo" command, described later.

Given this basic system setup, we present the basic feature set of Maestro.

4.3 Maestro Features

At first glance, Maestro's presentations appear very familiar: They consist of slides, bullet hierarchies, figures and other structures which are similar in form and function to those of contemporary presentation systems. Perhaps the only visual indication that Maestro is different from other systems is the presence of the staging area to the left of the slide (figure 2).

The staging area affords the system's two most basic commands, "next slide" and "previous slide". To move to the next slide, presenters place one hand in the center of the staging area, and move the hand straight down (figure 3a). Likewise, to move to the previous slides, presenters need only move their hand straight up, again starting from the center of the stage (figure 3b).

Relatively unique to Maestro is the ability to navigate *within* slides. Maestro allows presenters to author slides whose content is longer than the height of the projection screen. In these cases, presenters can vertically scroll their slides. To scroll down, the presenter places both hands in the stage's center region, and then moves one of the hands straight down (figure 3d). 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 3c).

Maestro affords direct interactation with the actual content of the slides. For example, in Maestro, blocks of text can be highlighted by pointing to them with one hand (figure 4a). Bullet lists can also be authored as hierarchies, with child points initially hidden. Later, at presentation time, the user can expand these hierarchies by pointing to the top-level bullet with one hand and performing a downward stroke with the other (figure 4d). Maestro's figures are equally interactive, allowing presenters to selectively enlarge figures embedded alongside text. When enlarged, the figures occupy the entire screen. To zoom into a figure, the presenter need only move both hands into the image, and then pull them apart, vertically (figure 4e).

Finally, Maestro allows presenters to open a "carousel" containing thumbnails of all slides in the presentation (figure 5). To access the carousel, the presenter places both hands in the stage's center section, and then pushes away from their body (figure 4f). The carousel occupies the space vacated by the slide. Using other gestures, the presenter is then able to randomly access any slide. Alternatively, she may choose to display any two slides side-by-side in a split-screen mode (figure 6). This allows presenters to directly compare and contrast the content of separate slides. Importantly, presenters can interact with each of the split-screen slides independently.

To make this system robust and usable, we focused on two research challenges: A continuous motion gesture language, and feedback mechanisms tuned to gesture-based interfaces. We turn to these issues next.

4.4 Gesture Language

Maestro's gesture recognition system must spot meaningful gestures in streams of continuous hand motion. More specifically, the system must *segment* the stream of input into potential gestures, then pass these segments into a gesture recognition system.

Maestro performs automatic gesture segmentation and recognition via three individual steps: Identifying an instantaneous cue demarcating the start of the gesture, recognizing the gesture's motion in space, and identifying a terminating cue demarcating the end of the gesture. In this system, the start and end cues are the most vital. We call these *segmentation cues* since they serve to uniquely segment gestures from the stream of input. Segmentation cues must be selected to prevent the accidental recognition of spurious gestures, while not being difficult to articulate by the user. With strong

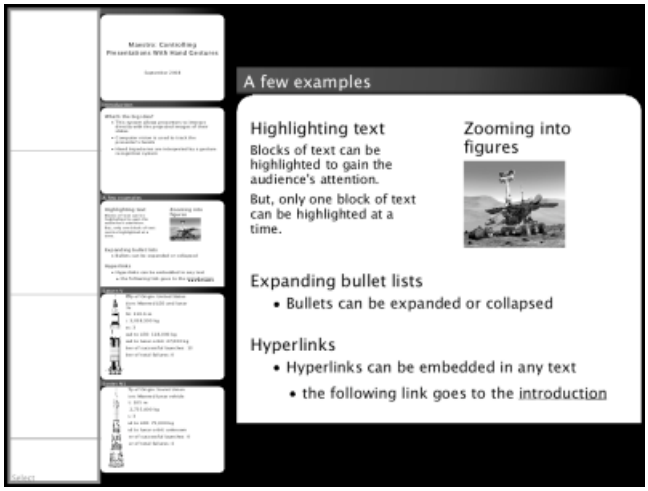

Figure 5: Maestro's carousel, which provides random access to any slide in the presentation.

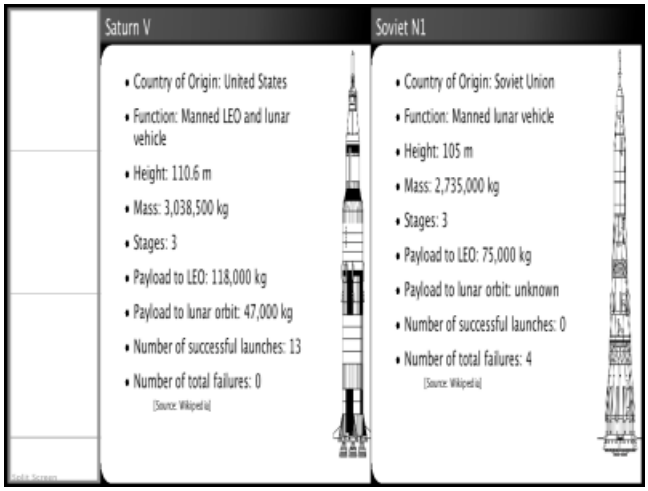

Figure 6: Maestro's split-screen mode, which displays two slides side-by-side.

segmentation cues, the actual gesture can more naturally map to the action to be performed, since it is the start and end cues that serve to segment the gesture from the stream of input, rather than gestures' unique paths in space. The motions of Maestro's gestures tend to be very direct and very linear. This affords the use of a very simple template-based gesture recognizer, where displacement (measured as distance and direction from the starting position), duration and path length are the salient features used to recognize trajectories.

Figures 3 and 4 show the gesture language of Maestro. While these figures clearly communicate gesture motion, they are less able to communicate the specific cues used for segmentation. A detailed discussion of these cues is thus warranted.

Maestro uses segmentation cues constructed from the following features:

- Spatial and content-based contextual features,
- Bimanual interaction,
- Hand roles, and
- Dwelling

While we list and later describe each feature-type separately, they are typically used in conjunction with one another to construct cues that are highly unlikely to occur by coincidence. For example, consider a cue requiring both hands to meet directly over a bullet point (as in the starting configuration of the hands in figure 4c and 4d). First, this cue requires that both hands rendezvous in space. Such an event is unlikely to occur by accident, but even if it did, it will not be considered a starting cue unless the point of rendezvous corresponds spatially with a bullet. Of course, an action will not fire unless there is a subsequent motion of the hand signalling the intent to invoke a command.

We now describe each of the segmentation feature-types individually.

4.4.1 Spatial and Content Context

To the extent possible, Maestro uses the content of the projected slides to contextualize the hand's motion. Our choice of spatial and motion features is motivated by so-called non-accidental features in computational perception [9]. Not only does spatial context provide excellent cues for demarcating the start- and end-points of a gesture, it also helps specify which objects the resulting command should operate on. Continuing the example above, the "expand bullet" gesture requires that both hands meet over a common bullet point. The staring cue for this gesture fully identifies *which* bullet point should be expanded. This is crucial for the development of a direct manipulation-style interface.

Gestures that operate on slides themselves present a particular challenge, as their spatial context is the entire display. Accordingly, Maestro's staging area creates a separate space to allow for manipulation of slides. This staging area thus introduces an artificial *spatial context*, so that spatial features can again be used in the construction of cues for segmenting and recognizing gestures. Moreover, because the staging area does not overlap with slide content, the derived spatial cues again fully specify the object of interaction – in this case, the slide itself.

4.4.2 Bimanual Interaction

Relatively unique to Maestro is the coordinated use of both hands to mark the start or end of a gesture. Simply stated, it is highly unlikely that two hands will accidentally *directly* meet in both space and in time. Maestro uses this fact to its advantage to assist with segmentation. A number of gestures require two hands to be initially close together (e.g., scrolling, zooming, and expanding / collapsing bullets), or together at the end of a gesture (e.g., in the case with zooming out).

4.4.3 Hand Roles

A *primary* role is assigned to the hand which least-recently entered the projected display area. Conversely, a *secondary* role is assigned to the hand which most-recently entered the display area. Roles may be reassigned as hands enter and leave, and are important when operating on objects that are densely packed within a region of the display. In such situations, the primary hand is used to specify the object of interaction. In this sense, the primary hand is used for fine-grained targeting.

While this notion of hand roles may seem to add complexity to the gesture language, we have found it rather natural in practice, since the user's dominant hand is already used to point at material on the slide. For these gestures, the second hand is simply pulled in to perfom an operation on the "selected" item. Maestro also provides feedback to remind users of these two roles. We describe all feedback mechanisms next.

4.4.4 Dwell time

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

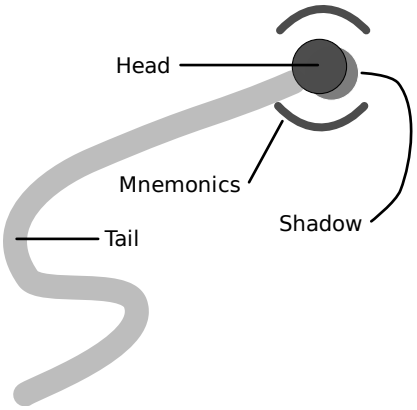

Figure 7: Structure of the hand tracking "comet". The *head* indicates the position to which Maestro believes the hand is pointing, the gesture *mnemonics* indicate which gestures can be performed in a given context, the *shadow* signals bimanual interaction, and the comet's *tail* indicates if the hand's speed is conducive to gesture recognition.

of a gesture in other systems. However, dwell time is used sparingly in Maestro, and mainly in those instances where the presenter is unlikely to be addressing the audience, such as when the presenter is selecting slide thumbnails from the carousel. The only exception to this is with hyperlinks; our user trials have shown that presenters immediately assume that they can follow a hyperlink by pointing, so our interface uses dwell-pointing for this purpose.

Our restricted use of dwell time was motivated by findings of our user study, described later. In particular, users found that the need to dwell interrupted the "flow" of giving a presentation. Accordingly, most gestures in Maestro avoid the use of dwell time to assist with segmentation.

4.5 Feedback and Affordances: "Feedback Comets"

The second major challenge we faced when designing Maestro was developing a mechanism for communicating command affordances and system feedback to the presenter. 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 system 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. This ensures that we do not interfere with the visual presentation of material.

Importantly, Maestro provides feedback before, during and after the performance of a gesture. In the latter case, Maestro acknowledges the receipt of some commands by displaying translucent icons in the staging area. We found these icons to be necessary because, when focusing on performing gestures in the stage, presenters frequently miss feedback displayed in other regions of the screen – even feedback as large as a slide transition. For pre-gesture and and in-gesture feedback, Maestro uses *feedback comets*.

Feedback comets are small cursor-like objects that follow the hands as they move around onscreen. As the name suggests, the comet metaphor relates the cursors to the astronomical object of the same name. Astronomical comets have a head nucleus, which is surrounded by a gas cloud called a coma, which is trailed by a dust tail. Maestro's feedback comet has a similar structure (figure 7).

At the most basic level, the comet's head nucleus reveals where the system *thinks* the presenter is pointing. While simple, we have found this basic feedback to be essential for learning how to use

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

Figure 8: Several gesture mnemonics used by Maestro. The dots in the "scroll up" and "scroll down" mnemonics indicate the presence of a stationary hand.

the system and understand its current state. Specifically, users can assess if the system is properly calibrated, can adjust for latency in the system, and can immediately recognize if the system stops responding. Other forms of feedback provided by the comet are discussed below.

4.5.1 Gesture mnemonics

Crucially, small gesture mnemonic icons are painted around the periphery of the comet, taking the role of the comet's dust coma. Gesture mnemonics are Maestro's primary form of pre-gesture feedback, serving both to indicate *which* commands are available in a particular context, and to remind users *how* to perform their gestures. For example, when the hand is placed in the staging area of a slide, the comet will be decorated with a mnemonic reminding the presenter that there is a previous slide that can be accessed by moving the hand upward. Importantly, 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. Mnemonics are not meant as detailed gesture instructions, but instead serve to indicate the direction and form of the gesture. A few of Maestro's mnemonics are listed in Figure 8.

4.5.2 Identity and role

Comets also provide feedback during the performance of a gesture. This *in-gesture* feedback communicates a significant amount of information about the gesture recognizer's internal state. In early system testing, we quickly realized that users benefitted from this exposure. For example, the comet's color identifies which hand the comet is following, *and / or* the role assigned to that hand. If the hand is assigned a primary role (as opposed to a secondary role), then the comet is rendered in color. Otherwise it is simply painted black. A red comet follows the red glove, and a blue comet follows the blue glove.

4.5.3 Speed

The length and width of the comet's tail communicates the appropriateness of the hand's speed for gesture recognition. A slow moving hand will have a short and wide tail, while a fast moving hand will have a long and thin tail. In this sense, the tail behaves as an "elastic," and stretches thin if moved quickly. If the hand moves too fast for the gesture recognizer, then the tail vanishes completely. Importantly, the comet tails are nearly transparent, and are very difficult to see from a distance; however, they are quite visible to the presenter.

4.5.4 Bimanual Indicator

Bimanual interaction is an important cue for segmenting and recognizing gestures, and the presence of two hands is a very strong indicator that a bimanual gesture is emminent. Occasionally, users intend on issuing a one-handed gesture, but mistakenly allow their other hand to be detected. Whenever two hands are detected, the tracking comet's nucleus gains a shadow to indicate that the system is tracking both hands. This type of feedback helps users "debug" the system when it does not respond as expected.

5 Evaluation and Lessons Learned

Maestro was developed through an iterative design process involving formative evaluations. Three volunteers were recruited to test an early exploratory implementation of the system, two of which were subsequently involved in testing all major and minor design iterations from that point onward, including a test of the final implantation. An additional three volunteers were recruited to test the final system in order to ensure that we were addressing the needs of new users. Generally speaking, all volunteers were able to reliably access all features of the system with very little training. They also found the feature set compelling. Most volunteers were particularly excited by the ability to scroll slides and zoom into images, and one volunteer was very impressed at the ease with which hyperlinks could be followed.

The user trials were each approximately 20 minutes in duration, and required that presenters give a mock presentation. The trails were videotaped, and an analysis of the final set of videos reveals that, out of 419 gestures attempted, 75% of attempts were detected correctly, 20% resulted in false-negatives, 3.8% resulted in false-positives, and the remaining 1.3% of attempts resulted in mode errors (a mode error occurs when a presenter confuses the role of their hands when performing a role-dependent gesture). These results can be compared to expert use of the system (by one of the paper authors), where an analysis of video reveals that, out of 197 gestures attempted, 91.3% were detected correctly, 7.6% resulted in false-negatives, and the remaining 1% resulted in false positives. There are a couple of points to be made here. First, we see an accuracy of approximately 75% for users who are learning to use a new system, with many of the false negatives a result of the subjects learning how to perform a gesture. These rates are similar to rates in other recognition-based systems. For example, Castelluci and MacKenzie [3] found initial correction rates of 26% and 43% for Graffiti and Unistroke input. Thus, initial use of our system is similar to other common recognition systems and, just as importantly, our expert data indicate that very high recognition rates are possible with training. Second, we note that the recognition errors are clearly skewed towards false negatives, for both novice and expert use. For this type of application, this is preferable as one interacts with the audience, false positives are clearly undesirable. This low false positive rate is not accidental, but the result of explicit decisions made in the gesture language.

5.1 Initial Gesture Language

While the final gesture language was found to be usable, the first gesture language employed was not. This first version used mainly one-handed gestures, and it did not incorporate a separate staging area for operating on the slides themselves. For example, to change to the next slide, one would sweep their hand along the lower edge of the slide from left-to-right. To return to the previous slide, one would sweep from right-to-left. The problem with these gestures was that they provided only weak cues for segmentation (namely, that the hand was somewhere along the bottom edge of the slide). This resulted in the preparatory motion between gestures being wrongly recognized as commands. For instance, to move to the previous slide, one would have to naturally reach over to the right side of the slide so that they could sweep from right-to-left. However, in doing so, the "reaching" would be recognized as the next slide gesture. Similar problems were observed with scrolling up and down, and expanding and collapsing bullets. While that particular gesture language likely would have worked well had we targeted a touch screen display, it was not workable in our vision-based system, where the symmetry between gestures caused problems.

We attempted to address this issue by requiring that all gesture begin with a short pause, essentially using dwell time to demarcate the start of every gesture. While this technique was rather effective from a gesture recognition point-of-view, users complained that the enforced pauses interrupted the natural flow of the presentation. To remedy this problem we experimented with reducing the duration of the dwelling. Unfortunately, this led to the "Midas touch" problem [8], where gestures may inadvertently be activated whenever, and wherever, the hands rest.

Finally, without the staging area, the early system had great difficulty differentiating between the gestures operating on slides and the gestures operating on the slide content. For example, both the scrolling gesture and expand bullet gesture incorporated a downward stoke. If, when scrolling, a user accidentally started over a bullet, the bullet would be expanded instead.

These observations led us to create strong segmentation cues, in particular, the use of two-hands and context to differentiate gestures. It also led us to explicitly *segment the space* with the staging area so that gestures targeted at the slide would not be confused with gestures targeted at elements *within* the slide. In the end, these modifications significantly improved the usability of the system, and made the interaction more fluid and less error-prone.

5.2 Maintaining the Illusion of a Touchscreen

When users first interact with Maestro, they approach the system as if it were a touchscreen display. We found this mental model had several implications for use of the system and how it is setup. Because users often conceptualize the system as if it were a touch-sensitive surface, they do not always consider the point-of-view of the camera and can thus orient themselves in ways that occlude the camera's view of the hands. If the occlusion causes the system to miss a gesture, users tend to focus on varying the speed and positioning of their hands, but overlook their orientation as a possible source of error. However, when reminded that the camera must have a clear view of the hands, users are much more careful of their posture. Noting this tendency, we ruled out numerous two-handed gestures that were found to encourage users to assume postures that could occlude the hands.

Maestro's hand tracking subsystem is designed to be robust to the geometries arising from a wide range of camera and projector placements. Unfortunately, for the same reasons described above, users forget about the camera and its placement. To prevent occluding postures, users must assume the camera's point-of-view. In our tests, we found that users form a mental model of occlusion that corresponds best to a camera placed directly in front of the screen, the same location of both the projector and the audience. This placement also creates shadows to correspond to areas not visible to the camera. With this placement, users naturally adopt postures conducive to recognition, because their primary task is to present ma-

terial on the slide in ways the audience (and hence, the system) can follow along.

5.3 Maintaining Recognition State to Provide User Feedback

When building a gesture recognition system, it is often tempting to use context and system state to rule out as many gestures as possible. This improves efficiency and helps to reduce false-positive rates. However, we found that one should not be too quick to rule out gesture possibilities because eliminating gestures "impossible" in a current context removes the possibility of providing feedback to the user. For example, one could argue that the "scroll down" gesture need not be considered when performing gestures within the context of slides that cannot be scrolled. However, presenters occasionally try to scroll such slides. By not considering this gesture, the system will not be able to inform the user that they are issuing an invalid command. This lack of system response is almost always attributed to a false-negative, and the user will repeat the gesture in error. Consequently, Maestro continues to recognize gestures that are invalid in a particular context so that it can provide appropriate feedback to the users.

5.4 "Undo"

Despite our best efforts to build a reliable system, failures may occur. These failures can originate from numerous sources including the vision-based hand tracker, the gesture recognizer, and even the presenters themselves. In the case of a false positive, a command is issued accidentally. Since such commands are unintentional, the presenter may not immediately recognize how the system state has changed. For example, if the "next command" is falsely detected, the presenter will probably notice that the slide has changed, but may be confused as to where in the slide deck the presentation has moved; the "previous slide" command or the accidental activation of a hyperlink would also result in a slide transition. For this reason, Maestro provides an "undo" command which can be accessed at any time. A label in the bottom-left corner of the staging area provides an indication of which command will be undone.

6 Conclusion and Future Work

In this paper we have introduced Maestro, a gesture-based presentation system which allows a direct manipulation-style of interaction with the contents of a projected slideshow. Maestro uses a web camera for input, and employs computer vision to detect and track the presenter's hands. Despite not having an explicit segmentation of the user's hand trajectories, Maestro is able to reliably recognize a wide range of gestures. This is achieved by using robust segmentation cues derived from non-accidental features of the observed hand motion. Importantly, segmentation cues allow the gestures to remain quite simple, and to better mirror the actions that are to be performed.

Additionally, Maestro introduces the *feedback comet*. The comet addresses the need to provide users with feedback, while not detracting from the visual appearance of the presentation. The feedback comet's primary role is to present gesture mnemonics to the user. These mnemonics serve to inform and remind the user of the gestures that can be performed in a given context. The feedback comet also reveals much of the gesture recognizer's internal state, allowing presenters to detect and prevent recognition errors whilst in the midst of performing a gesture.

There are numerous improvements that can be made to the existing system, and there are equally many opportunities for future research. In particular, we would like to improve the hand tracking system to allow presenters to interact with the system without the need for colored gloves. Research by Hilario *et al.* has made great strides in this direction [6]. Provided that bare hands hands can be detected and tracked effectively, Maestro's gesture language should continue to work unmodified; the gesture language treats each hand interchangeably.

Finally, the feature set of Maestro enables a qualitatively different way of giving a presentation, one in which the presenter has more direct access to the projected content. At the same time, Maestro introduces features not commonly found in popular presentation systems, such as the ability to scroll slides or show a split-screen view of two slides. While we have evaluated the basic usability of Maestro, there is an open question as to how these interaction mechanisms and feature sets can lead to styles of presentation not easily achieved with current systems. Thus, we would like to perform longitudinal studies of this system to understand how its characteristics influence presentations, both in terms of content and delivery.

References

[1] T. Baudel and M. Beaudouin-Lafon. Charade: remote control of objects using free-hand gestures. *Commun. ACM*, 36(7):28–35, 1993.

[2] X. Cao, E. Ofek, and D. 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.

[3] S. J. Castellucci and I. S. MacKenzie. Graffiti vs. unistrokes: an empirical comparison. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 305–308, New York, NY, USA, 2008. ACM.

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

[5] Google. Tech talks at google. `http://research.google.com/video.html`, September 2008.

[6] M. N. Hilario and J. R. Cooperstock. Occlusion detection for front-projected interactive displays. In *Second International Conference on Pervasive Computing*, Linz/Vienna, Austria, 2004. Springer Berlin Heidelberg.

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

[8] R. 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.

[9] 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).

[10] H.-K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(10):961–973, 1999.

[11] L. Mamykina, E. Mynatt, and M. A. Terry. Time aura: Interfaces for pacing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 144–151. ACM Press, 2001.

[12] L. Nelson, S. Ichimura, E. R. Pedersen, and L. 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.

[13] Palm, Inc. Graffiti2 tips. `http://www.palm.com/us/support/handbooks/graffiti2_tips.pdf`, September 2008.

[14] I. Parker. Absolute powerpoint: Can a software package edit our thoughts? *The New Yorker*, Annals of Business Section:76–87, May 28 2001.

[15] SMART Technologies. Smart board interactive whiteboards. `http://www2.smarttech.com/st/en-US/Products/SMART+Boards/default.htm`, September 2008.

[16] C. von Hardenberg and F. 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.