Factored HMMs for Bimanual, Context-Dependent Gestures

Adam Fourney
afourney@cs.uwaterloo.ca

Richard Mann
mannr@uwaterloo.ca

Michael Terry
mterry@cs.uwaterloo.ca

David R. Cheriton School of Computer Science, University of Waterloo
Technical Report CS-2010-09

Abstract

As we expand our use of hand gestures for interacting with computational devices, the spatial context in which gestures are performed becomes an increasingly important feature for interpreting user intent. In this paper, we demonstrate how spatial context, and bimanual coordinated hand motion, can be efficiently modeled using a factored hidden Markov model. This factorization, guided by topological constraints relating to feature extraction, reduces the number of modeling parameters by two orders of magnitude compared to an unfactored model. We used these factored models when constructing a gesture-based presentation system called Maestro*. We then performed a series of experiments to evaluate the performance of Maestro's recognizer with and without the spatial and bimanual context features.*

1. Introduction

Gesture-based interaction is quickly becoming a realistic input modality for a variety of computing devices and applications. Hidden Markov models have emerged as a means to constructing robust and efficient gesture-recognizers [5, 9, 14, 18, 19]. In this paper, we consider the problem of efficiently representing context (spatial and motion) in a discrete hidden Markov model (DHMM).

Our research is motivated by our work in developing *Maestro*, a gesture-based presentation system. Electronic presentations are one of the most frequently proposed application spaces for this type of interaction because people naturally gesture at slides when giving presentations. To inform the design of a gesture-based presentation system, we performed observational studies, noting how people gesture when giving technical presentations [7]. We observed that presenters often gesture by pointing to specific landmarks on the screen (single hand spatial context), by positioning both hands to demarcate an object (bimanual coordinated position), or by moving both hands (bimanual coordinated motion). The design of Maestro's single hand gestures was further informed by our previous work in [6], where we explored the properties of this class of gesture; this research uncovered tight clustering for various motion regularities (e.g., stopping events, vertical motion, horizontal motion, etc.) Both the observational study, and the aforementioned research, helped directly inform the design of Maestro's gesture language.

In [7] we presented an overview of Maestro from the perspective of human computer interaction (HCI). There we focused on the design of the interface and interaction. We also reported the results of a performance evaluation where Maestro was used for lecturing to undergraduate university classes.

In this paper, we focus on the details of Maestro's gesture recognition component. In particular we show how bimanual and spatial context can be encoded efficiently in a discrete hidden Markov model with a factored observation model. A factored observation model is required due to the large number of features we consider. A naive implementation would directly model the joint distribution of all features (positions and velocities of both hands, etc.). However, since several of the features are conditionally independent, we can significantly reduce the number of parameters required.

Maestro's DHMMs are factored using a few simple topological constraints. A simple example is the relative position of the two hands, or of one hand to a landmark. In this case, the relative position is a function of the position of both objects, and can be taken to be independent of the velocities of those objects. Similarly, when observations are missing, such as lost position or velocity measurements, all features derived from the measurements are taken to be missing. The above are general topological constraints that depend on the feature types only, and do not change the potential dependency assumptions among the objects. Nonetheless, we will show that enforcing these constraints

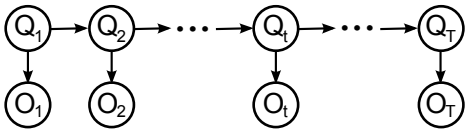


Figure 1: The dynamic Bayesian network representation of a hidden Markov model.

significantly reduces the number of parameters in our gesture models.

The first two parts of this paper review related work, and describe Maestro's gesture language and gesture recognition system. The third part of the paper presents a simple set of topological constraints, and shows a graphical model for the observations. The fourth part of the paper shows the performance of the recognizer, and evaluates our system with and without the spatial and bimanual context features. The paper concludes with a discussion of future research.

2. Background

In this section we review discrete hidden Markov models, and we describe previous work on using DHMMs to model contextualized gestures. Following this discussion, we describe the details of Maestro's gesture recognizer in sections 3 and 4.

2.1 Discrete hidden Markov models

Hidden Markov models (HMMs) are graphical models used for modeling sequential observations, such as the evolution of signals over time. *Discrete* hidden Markov models (DHMMs) are HMMs operating over a discrete alphabet of output symbols (observations). Formally, DHMMs adhere to the dynamic Bayesian network depicted in figure 1, 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 $\mathbf{S}$.
- 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, not on the value of t. In other words, the state transitions follow a *homogenous* Markov process.
- An 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.

It is important to keep in mind the properties that give rise to the DHMM's namesake; first, the state sequence evolves according to a 1st-order homogeneous 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. As such, the state sequence must be inferred from the observations.

2.2 Related Work

HMMs have been used extensively for hand gesture recognition [5, 9, 14, 18, 19]; however, comparatively little work describes how best to model spatial relations in this framework. In many cases, observations are intentionally abstracted in order to achieve various levels of translation and rotation invariance. For example, translation invariance is often achieved by taking the hand's instantaneous velocities as features rather than positions [2, 11]. Alternatively, it is common to compute features based purely on the direction of hand motion [4, 9, 12]. These approaches work quite well for recognizing semaphoric gestures (similar to those found in the US army field manual "FM 21-60: Visual Signals" [1]). Unfortunately, there are many practical scenarios where the meaning of a gesture partially depends on the spatial context in which it is performed. For example, this is the case with many signs in American sign language (ASL) [16], and also with gesture-based interfaces seeking to replicate the direct manipulation style of interaction common to today's graphical user interfaces.

There have been numerous approaches to adding or representing spatial relations in HMM gesture models. Perhaps the simplest approach is to simply model the hand position directly. Unfortunately, this approach is not well founded; in an HMM, observations depend only on the state in which the machine finds itself in at a given moment in time. Importantly, these state-specific observation distributions do not evolve with time – making them inappropriate for modeling even the simplest gestures where the hands' positions are changing under constant non-zero velocity (e.g., moving straight down). One way to address this challenge is to coarsely divide space into large regions, and to model the hands' paths through these regions rather than modeling exact positions [19]. Unfortunately, this coarse approach throws away much information regarding the shape of the gestures. This shape information is very valuable for differentiating between many gestures.

Okumura *et al.* , seeking to build a system for recognizing Chinese characters in handwriting, proposed a modified HMM which addresses the aforementioned problems. In

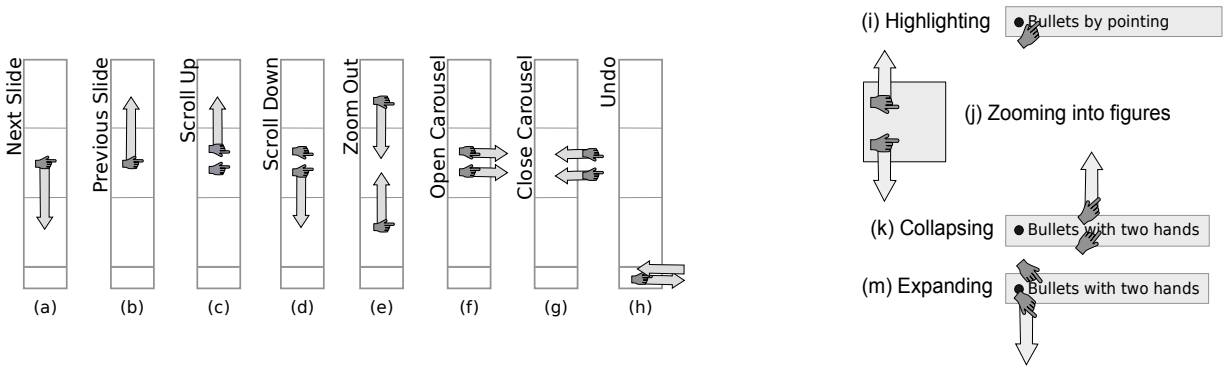


Figure 2: Maestro's navigation gestures (left), and content gestures (right).

their system, position and direction features are used interchangeably depending on how each of the HMM's states is entered; direction is used as the observation when a state is reentered following a self-transition, and position is used when a state is entered by an inter-state transition. The authors reported much success using this approach. We attempted to adapt this approach to Maestro, but found that the approach did not place enough weight on the position observations for this context information to meaningfully guide inference; our models and motion sequences made far more use self-transitions as compared to inter-state transitions.

Wilson *et al.* augment HMMs with global parameter(s) to create parametric HMMs (PHMMs). While this approach allows global properties such as scale or reference points, we require features (particularly, relational features), which are state dependent.

Finally, the discussion has thus far considered spatial context as capturing details regarding the hand's motion with respect to a landmark or a region of interest. An alternative form of context comes from the coordinated motion of the presenter's hands when performing bimanual gestures. Relatively little literature discusses how to model bimanual gestures. Siskind and Morris [15] used relational features to recognize simple motion verbs. While this system was successful, no attempt was made to factorize the spatial and velocity distributions. Brand *et al.* present a "coupled hidden Markov model" (CHMM), consisting of multiple HMMs (e.g., one for each hand). The CHMM conditions the state transitions for each HMM to consider both the current state as well as the states of the other HMMs to which it is coupled [3]. While this approach is a mathematically elegant way to model loosely interacting processes, the flexibility comes at a cost of increased complexity (and training sample sizes).

In the remainder of this paper we will present a compact and simple approach to modeling spatial context and coordinated hand motion in a DHMM. We begin the discussion by describing Maestro's gestures, and we provide an overview of its gesture recognition machinery. We then discuss the DHMMs in more details, followed by an evaluation of our approach on real-world data.

3. Gestures in Maestro

Maestro was developed with the expressed purpose of quickly and inexpensively exploring the implications of gesture-based interactions with presentations. This system employs the use of a single web camera and is particularly simple; hands are detected and tracked 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. The tracking system reports on the positions of both gloves at a rate of 15 times per second. Although both gloves occupy a sizeable 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 [8]). The input to Maestro's gesture recognition system thus consists of a pair of point trajectories. We extract features from these trajectories which are then modelled by a DHMM. In the sections that follow we present an overview of Maestro's gesture language and we describe the DHMMs used by Maestro.

3.1 Gesture language

Maestro allows presenters to use hand gestures to both navigate the slide deck (e.g., to advance slides), and to interact directly with the *content* of their slides (e.g., to zoom into figures, or to expand bullet hierarchies). We now describe these navigation gestures (figure 2, left) and content gestures (figure 2, right) in more detail.

3.1.1 Presentation navigation

Maestro's navigation gestures allow presenters to move between slides, to scroll slides, and to bring up the slide carousel. These gestures are independent of slide content, and are thus performed in the left margin of each slide, a region we call the *staging area* (see left side of figure 2). 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 2a). 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 2b). A set of horizontal ruled lines delineates the areas for invoking these gestures, but these visual guides appear only when the presenter rests their hand within the margin for a short period of time. Gestures can be performed even when the guidelines are not visible.

Unique to Maestro is the ability to scroll up and down *within* slides. Content can be scrolled by placing both hands in the stage's center region, and then moving one of the hands straight down (figure 2d). 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.

Finally, Maestro 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 2f). Using other gestures, the presenter is then able to randomly access any slide.

3.1.2 Interactions with slide content

Maestro also affords gestural interaction with the actual content of the slides (see right side of figure 2). This class of gestures is particularly context sensitive. Blocks of text can be highlighted by pointing to them with one hand. Presenters can also selectively enlarge figures embedded alongside text. When enlarged, a figure occupies the entire screen. To zoom into a figure, the presenter moves both hands into the figure, then pulls them apart vertically (figure 2j).

Finally, presenters can also author slides with hierarchical lists of bullets, with child bullets initially hidden. To reveal child bullets, the presenter places both hands next to the bullet point of interest, and slides one hand down, similar to the scroll gesture (figure 2m). The reverse motion hides the child bullet point.

3.2 Modeling gestures

Each of Maestro's gestures is modeled independently as a DHMM. The general definition of a DHMM from Section 2.1 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). 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 [9, 13, 14, 19] where state s_i is connected to state s_j only if $0 \leq j - i \leq \Delta$.

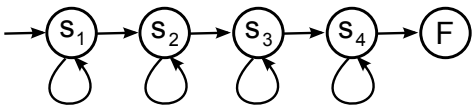


Figure 3: The topology of the 4-state DHMMs used by Maestro.

In Maestro, all gesture models take the form of a 4-state DHMM where $\Delta = 1$. This topology is depicted in figure 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.

3.3 Gesture spotting

Maestro must be able to recognize meaningful hand gestures that are embedded in sequences which also contain non-gesture (background) hand motion (e.g., to account for a presenter's gesticulation). 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. To accomplish gesture spotting, we arrange the individual gesture DHMMs into a "gesture spotting network", as described by Lee *et al.* in [9]. This gesture spotting network, depicted in figure 4, is itself a discrete hidden Markov model. It is constructed by connecting the individual gesture models in parallel. Additional "garbage" or "filler" models [5, 17] are added to directly model the background process. 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. Time-synchronous Viterbi decoding is then 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.

Maestro's filler models consist of a one-state "silence" model, to account for sequences in which neither hand is detected; and a "catch-all" model, which accounts for any additional hand motion. We use the catch-all model sug-

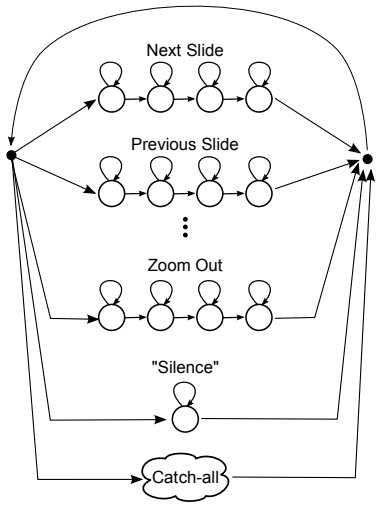


Figure 4: Maestro's gesture spotting network.

gested by Lee *et al.* in [9]. The catch-all model simply explains all hand motion by an arbitrary ordering of piecewise linear motion segments in any direction. While all hand motion can be explained by a path through the catch-all model, sequences consisting of gestures will prefer a Viterbi path leading through the more specialized gesture models.

4. Features and a factored observation distribution

In this section we present the set of features which Maestro uses to model gestures, along with an efficient factorization of the joint observation distribution. Importantly, the observation distribution is factored using conditional independence relations that are guided by topological constraints.

4.1 Gesture features

Similar to previous research [4, 9, 12] Maestro uses a measure known as "direction" or "turning angle" as a stable feature for modeling gestures. Consider the pair of sequential observations $\vec{X}_{t-1}^{(R)}$, $\vec{X}_t^{(R)}$ of the red glove's position at times $t-1$ and t, respectively. The turning angle $\theta_t^{(R)}$ is defined as the angular component of the finite difference $\vec{X}_t^{(R)} - \vec{X}_{t-1}^{(R)}$, when expressed in polar coordinates. 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.

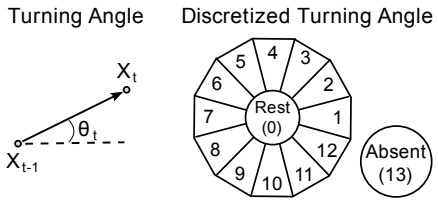


Figure 5: The discrete turning angle feature used for modeling gestures.

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 5).

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 $||\vec{X}_t^{(R)} - \vec{X}_{t-1}^{(R)}||_2$ in order to determine when the glove is considered to be "near rest".

4.2 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 section 3, 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).
- 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.

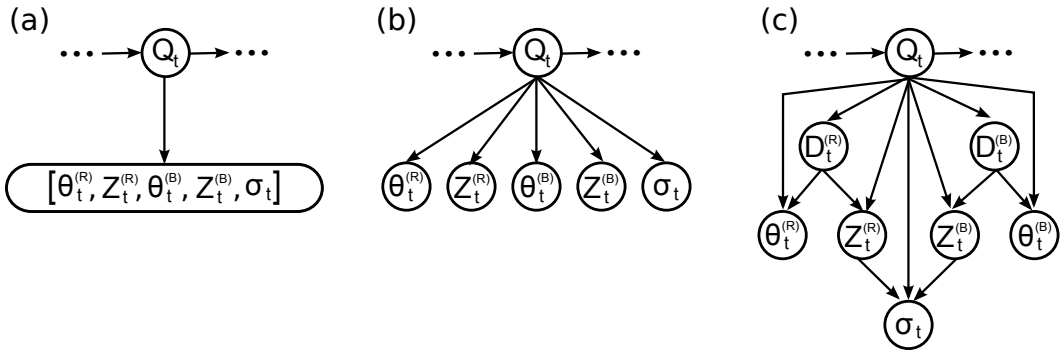


Figure 6: Various factorizations of the observation distribution.

4.3 Spatial relation between hands

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 $\vec{X}_t^{(R)} - \vec{X}_t^{(B)}$ rather than $\vec{X}_t^{(R)} - \vec{X}_{t-1}^{(R)}$. 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 $||\vec{X}_t^{(R)} - \vec{X}_t^{(B)}||_2$ is small. These short vectors indicate that the hands are "together".

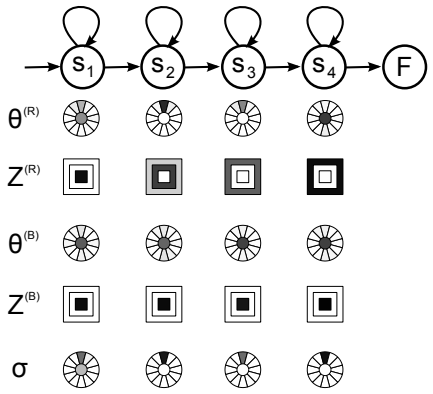
4.4 An efficient factorization

Together, the aforementioned features are represented by the discrete feature vector $\vec{f} = [\Theta_t^{(R)}, Z_t^{(R)}, \Theta_t^{(B)}, Z_t^{(B)}, \sigma_t]^T$. In this environment, the most straightforward approach to modeling a DHMM state's observation distribution is to represent it directly using a histogram. The corresponding Bayesian network is depicted in figure 6a. In this case, the histogram would consist of one bin for every possible feature vector $\vec{f}$, and each bin would contain the probability $P(O_t = \vec{f}|Q_t)$. Unfortunately, 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.

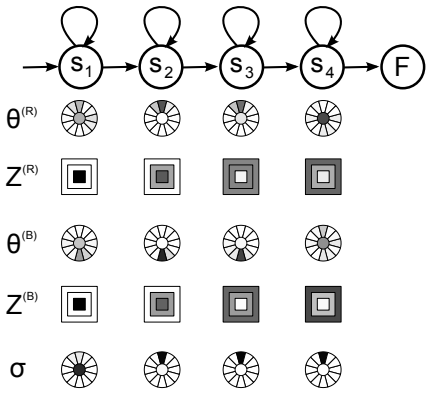
One way to resolve this issue is to assume that each of the feature vector's components are conditionally independent given the HMM state (as in figure 6b). This reduces the number of parameters to 50 for each state – an immense savings! However, such a factorization is unwarranted since the various components of $\vec{f}_t$ are not conditionally independent. In addition to omitting dependencies, independence assumptions may admit inconsistent states. For example, 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 it is impossible for the feature σ_t to take on a value indicating that the hands are close together.

We use these topological constraints to factor the observation distributions $P(\vec{f}_t|Q_t)$ according to the Bayesian network depicted in figure 6c. Our factorization includes dependencies among all spatial context features $(Z_t^{(R)}, Z_t^{(B)}, \sigma_t)$. Furthermore, since σ_t depends only on the position features, we avoid dependencies on the motions $\Theta_t^{(R)}$, and $\Theta_t^{(B)}$. Note that, for each of the aforementioned cases, the conditional dependencies follow directly from the manner in which features are computed. In this sense, we call these relationships "topological constraints". This can be contrasted with other conditional independencies which are assumed solely for the purpose of simplifying the model. For example, an HMM assumes states evolve according to Markov process, where the next state is conditionally independent on the past history of states, provided that the current state is known. This Markovian assumption is probably not entirely accurate, but significantly simplifies modeling.

Notice 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. Importantly, they allow us to account for cases where the hands are not detected while preserving conditional independence between the position and direction features. The resulting



(a) Visualization of the "scroll up" DHMM. (Gesture motion depicted in figure 2c)



(b) Visualization of the "Zoom In" DHMM. (Gesture motion depicted in figure 2j)

Figure 7: In this figure, each cell represents a possible value for the discrete turning angle ($\Theta^{(R)}$ and $\Theta^{(B)}$), zone ($Z^{(R)}$ and $Z^{(B)}$), and spatial relation (σ) features. The intensity of each cell represents the *marginal probability* of observing the value for the corresponding feature.

factored observation distribution requires only 300 parameters.

5. Evaluation of Maestro's contextual features

In this section, we present a set of formal experiments used to evaluate Maestro's gesture recognizer in a series of gesture spotting tasks. These experiments quantitatively establish the importance of the contextual features used in our models. The discussion begins with a description of the procedure used to train the models. We then describe the results of the gesture spotting experiments in more detail.

5.1 Training and parameter estimation

As noted earlier, Maestro models each gesture with a 4-state left-right hidden Markov model consisting of approximately 300 parameters per state. In order to train these DHMMs, we gathered approximately 100 isolated examples of each of Maestro's gestures. Since there are 11 gestures, a total of 1,100 training examples were collected. In the case of context-sensitive gestures, such as expand, collapse, and zoom-in, training was conducted by randomly relocating the landmarks after each gesture performance. This avoids learning a model that is specific to a single location or slide layout.

Having acquired the training data, model parameters were learned using the Baum-Welch reestimation procedure. Since the training data represents isolated gestures, five-fold cross validation can be used to evaluate the performance of the DHMMs in an isolated gesture recognition task. While isolated gesture recognition is arguably simpler than gesture spotting, the results of this experiment were used as an initial "sanity check" to verify that the training was proceeding as expected. This experiment revealed that between 98% and 99% of the isolated gestures were recognized correctly across each of the five folds. These positive results suggest that the models are able to accurately discriminate Maestro's gestures from one another.

In addition to performing cross validation, an inspection of the resulting models reveals that the reestimation procedure seems to have well-captured the essence of each gesture. As an example, the "scroll up" gesture's DHMM is depicted graphically in figure 7(a), and the "zoom in" gesture's DHMM is depicted in figure 7(b).

5.2 Method

Motivated by the initial positive results of the isolated gesture recognition task, a set of formal experiments was conducted to evaluate the models when used for gesture spotting. These experiments numerically establish the importance of the contextual features used in our models. To accomplish this, we compared the gesture spotter's error rates across numerous models utilizing different subsets of the contextual features. Specifically, in the first experiment, the full set of features was utilized; in the second experiment, the spatial relation feature σ_t was dropped from the model; in the third experiment, the zone features $Z_t^{(R)}$ and $Z_t^{(B)}$ were dropped; and in the fourth experiment, all three contextual features were dropped (leaving only the turning angle features).

In order to compare gesture spotting rates across the four experimental conditions, we video recorded the performance of 10 instances of each gesture. The video was manually coded to establish a ground truth for the tim-

Condition	Correct	False Negatives	False Positives
1. Full Model	107	3	4
2. Without σ	100	10	8
3. Without $Z^{(R)}$, and $Z^{(B)}$	91	19	56
4. Without σ, $Z^{(R)}$, and $Z^{(B)}$	88	22	82

Table 1: Recognition rates for each of the four experimental conditions.

ing of each gesture performance. However, the video was presented to the gesture recognizer as one continuous sequence. As such, hand trajectories included both gesture and non-gesture hand motion; gestures would have to be spotted. This data was then used as input in each of the four experimental conditions. To ensure results were directly comparable across conditions, a standardized set of presentation slides provided the contextual information with which the gestures were interpreted. While the standardized slides resembled a typical presentation, the slides were static and did not respond to gestures. As such, the context in which gestures were performed remained constant even as the rates of false positives or false negatives varied.

5.3 Results

The results from the four experiments are presented in table 1. In this table, we count the frequency of correct gesture spottings as well as the number of recognition errors. In this work we count both false negative errors and false positive errors. A false negative is counted when a gesture is performed by the presenter but no gesture is spotted by the recognizer. False positives are counted when the recognizer spots a gesture that does not correspond to any gesture performance. It is also possible for one gesture to be mistaken for another (e.g., the "next slide" gesture mistaken as the "previous slide" gesture), but this did not occur in our data (leading to a fully diagonal confusion matrix).

As expected, both the number of false negatives and the number of false positives increase as contextual features are removed from the model. Here, the zone features have the largest impact, especially in regards to false positives. This is because, without the zone features, gestures can be recognized anywhere in space; this leads to a large number of gesture detections during the preparatory motion that occurs between gestures. For example, the "previous slide" gesture might be mistakenly detected when the presenter initially moves their hand up to interact with the presentation. While the zone features are essential for a robust system, the spatial relation feature also appears to provide some inferential leverage to further reduce the number of errors.

6. Discussion

In this paper we have shown how spatial context and coordinated hand motion can be efficiently modeled in a DHMM for the purpose of building a gesture recognizer. This was demonstrated by selecting a set of discrete features describing the hands' positions with respect to a common region of interest (landmark), as well as a feature describing the the hands' positions with respect to one another. Factorization was achieved by recognizing various conditional independencies that follow directly from topological constraints based on the way in which features are computed. This factorization reduced the number of parameters by two orders of magnitude.

Additionally, we evaluated our models, and the context features, experimentally. As expected, the gesture recognizer was most accurate when using the full set of contextual features; accuracy dropped for each feature that was removed. Additionally, these experiments clearly demonstrate the importance of the spatial "zone" features, which communicate the locations of the presenter's hands. Possible future work would be to extend these results to mixed discrete-continuous hidden Markov models, where the direction and spatial relation features are modeled using a mixture of Gaussians rather than a simple histogram. Additionally, we will consider other topological constraints related to instantaneous changes in motion [10].

References

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

[2] 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*, Washington, DC, USA, 1998. IEEE Computer Society.

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:994, 1997.

[4] A. Croitoru, P. Agouris, and A. 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.

[5] 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.

[6] A. Fourney and R. Mann. Non-accidental features for gesture spotting. *Computer and Robot Vision, Canadian Conference*, 0:116–123, 2009.

[7] A. Fourney, M. Terry, and R. Mann. Understanding the effects and implications of gesture-based interaction for dynamic presentations. Technical Report CS-2010-03, David R. Cheriton School of Computer Science, University of Waterloo, February 2010.

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

[9] 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.

[10] R. Mann and A. D. Jepson. Detection and classification of motion boundaries. In *Eighteenth national conference on Artificial intelligence*, pages 764–769, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[11] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama. Substroke approach to hmm-based on-line kanji handwriting recognition. *Document Analysis and Recognition, International Conference on*, 0:0491, 2001.

[12] 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.

[13] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[14] G. Rigoll, A. Kosmala, and S. 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.

[15] J. M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *ECCV96*, pages II:347–360, April 1996.

[16] R. A. Tennant and M. G. Brown. *The American Sign Language handshape dictionary*. Gallaudet University Press, 1998.

[17] J. Wilpon, L. Rabiner, C.-H. Lee, and E. 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.

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

[19] J. Yang and Y. Xu. Hidden markov model for gesture recognition. Technical Report CMU-RI-TR-94-10, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1994.