Non-accidental features for gesture spotting

Adam Fourney
afourney@cs.uwaterloo.ca

Richard Mann
mannr@uwaterloo.ca

David R. Cheriton School of Computer Science
University of Waterloo

Abstract

In this paper we argue that gestures based on non-accidental motion features can be reliably detected amongst unconstrained background motion. Specifically, we demonstrate that humans can perform non-accidental motions with high accuracy, and that these trajectories can be extracted from video with sufficient accuracy to reliably distinguish them from the background motion. We demonstrate this by learning Gaussian mixture models of the features associated with gesture. Non-accidental features result in compact, heavily-weighted, mixture component distributions. We demonstrate reliable detection by using the mixture models to discriminate non-accidental features from the background.

1 Introduction

For many years, researchers have explored the visual perception and recognition of hand gestures as a mechanism for interacting with computers. One of the primary challenges faced when developing vision-based gestural interfaces, is that of gesture segmentation [11, 10, 3]. The issue arises because cameras stream observations including both gesture and non-gesture motion. A gestural interface must be able to "spot" meaningful gestures in these longer motion sequences. We encountered this issue when developing *Maestro* [5], a gesture-based presentation system. To address the segmentation issue, Maestro introduced a set of cues to signal the start or end of a gesture. For example, several of Maestro's gestures require that users begin by placing both hands together directly over a specific image location. Informally, these cues were selected to be easy to perform and to detect, and yet be unlikely to occur by accident.

Maestro's segmentation cues are motivated by so-called *non-accidental features* in computational perception. Researchers have long advocated the use of non-accidental features for image interpretation [12, 7]. The premise is that image features, such as parallel, collinear, and coterminating edges provide strong evidence for regularities in the world. For example, coterminating edges in the image are likely to arise from coterminating lines in the world (eg., due to corners or occluding boundaries of objects), collinear edges are likely to arise from collinear lines in the world, and so on. Similar arguments may be made for motion features, such as a rigid collection of moving points [18].

Most gesture recognition work has focused on achieving robust recognition for general gestures, such as arbitrary strokes, repeated (waving) gestures, and even sign language. Robustness is achieved through time warping (eg., *condensation* [2]), or explicit state-based models (eg., *hidden Markov models* [20, 3, 11]). In this paper we argue that choosing gestures based on non-accidental features allows for reliable gesture spotting. Specifically, we claim that humans can perform non-accidental motions with high accuracy, and that these trajectories can be extracted from video with sufficient accuracy to reliably distinguish them from the background motion. Our approach is similar to [16] except their work focused on character strokes in written language; instead, we ask if there are non-accidental features in gesture. We analyze gestures by learning Gaussian mixtures models of the features associated with gesture. Non-accidental features result in compact, heavily-weighted, mixture components. We demonstrate reliable detection by using the mixture models to discriminate non-accidental features from the background.

The remainder of this paper will discuss each of these steps in detail. The discussion begins with a motivating example.

2 Maestro - a motivating example

Gesture-based control of PowerPoint presentations is often cited as an example of hand gestures in human computer interaction research. This has led to the implementation of numerous demonstration systems [11, 1, 19]. Typically, these systems graft gestural control onto existing presentation software not originally designed with gesture control

978-0-7695-3651-4/09 $25.00 © 2009 IEEE
DOI 10.1109/CRV.2009.16

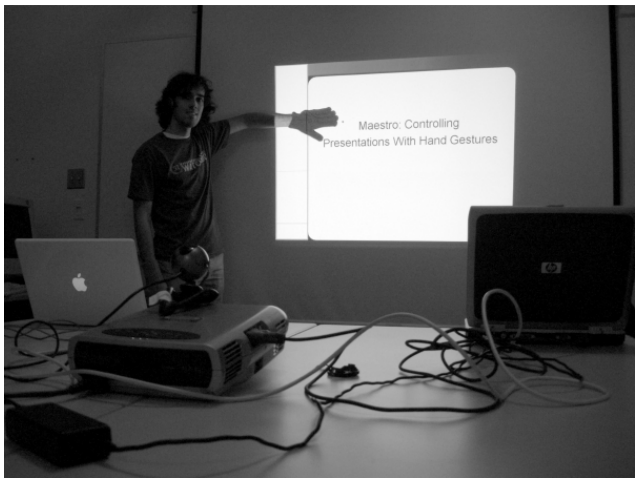

Figure 1: A user controlling the Maestro presentation system using only hand gestures

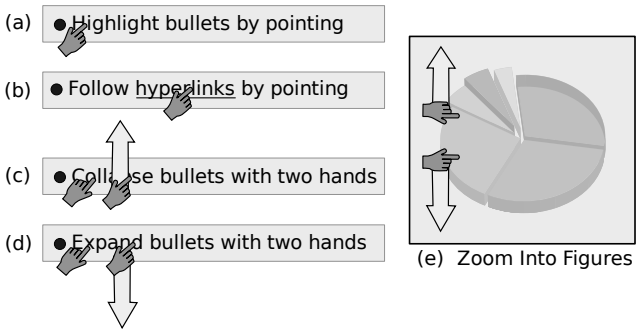

Figure 2: A few examples of gestures in Maestro. Note that in (c) and (d), the start of the gesture is signalled by the observation of both hands together over a bullet point. In (e), the gesture is signalled by a similar observation.

inmind. We developed *Maestro* to explore the unique challenges and opportunities that arise from incorporating gestural control from the ground up [5]. For example, Maestro allows presenters to interact directly with the *content* of projected slides (figure 1), while the aforementioned presentation systems typically limit gestures to a few navigational commands (such as those used to move *between* slides). Some of Maestro's gestures are listed in figure 2.

Maestro uses a set of cues to signal the start or end of a gesture. This helps Maestro "spot" gestures in long sequences of hand motion. Most of these cues depend upon the motion of the hands and/or on the spatial context in which the motion is observed. In the case of spatial cues, the content of the projected slides provides the necessary context. For example, a cue might require that a user place their hand near a "bullet" symbol when interacting with an item from a bulleted list. Additionally, many of these cues require the coordinated use of both hands. For example, to signal the start of the "expand" or "collapse" gestures (figures 2d and 2c) both hands must appear together over a bullet point.

3 Gestures inspired by non-accidental features

We would like to define a set of gestures whose performance gives rise to non-accidental features in the observed hand motion. This is guided by the hypothesis that such gestures will be easily differentiated from background motion. This requires a more formal description for non-accidental features and so we refer to work done by Jepson and Richards who provide a Bayesian justification of non-accidental features [8]. In their work, they describe the world as having various properties that occur probabilistically. These properties cannot be observed directly, but can only be inferred from various features that arise from observations. Any number of features can be derived from the observations, but we limit our discussion to features which support the *reliable inference* of world properties. Let F denote a world property and f denote the observed feature. The inference is reliable if, and only if:

1. $P(F) > 0$
2. $\frac{P(f|F)}{P(f|\neg F)} >> 1$

In other words, the inference is reliable if the properties have nonzero prior probability (1), and the likelihood ratio of the feature given the property is large (2). Essential to the non-accidental feature argument is the idea that, in the absence of measurement noise, non-accidental features occupy a lower dimensional subset of the feature space. These subsets correspond to so-called "world regularities", and ensure that the aforementioned likelihood ratio increases without bound as the measurement error decreases. For example, if we consider line segments in the plane, non-accidental features could include various low dimensional subspaces, such as parallel, collinear, or coterminating lines, right angles, midpoints of lines, etc. [4].

Designing a gesture language to yield non-accidental features requires some assumptions about the types of regularities that a person can reliably introduce when instructed. For example, a person may be able to reliably move their hands (more-or-less) straight down, but will probably not achieve the same level of reliability when instructed to move at a precise angle of 53^o. Similarly, a person may be reliable when instructed to "stop", but not when instructed to move at some other specific velocity.

Our work begins by assuming the following regularities: horizontal movement, vertical movement, and rest. Vertical movement is defined as any motion where the hand's vertical position is allowed to vary with time, while the horizontal position is held fixed. In this sense, vertical motion has 1

degree of freedom. Horizontal motion is defined similarly, and also has one degree of freedom. The category "rest", fixes both the hand's horizontal and vertical positions, and has no degrees of freedom. These categories are in contrast to "unconstrained" motion where neither the horizontal nor the vertical positions are fixed.

We consider gestures consisting of two motion segments, each of which can take on one of the four aforementioned categories. In total, each gesture allows up to 4 degrees of freedom (each segment contributes 2 degrees). Following [4] we can draw a *category lattice* based on the subspace relations of the categories (see Fig. 3). The degrees of freedom (or *dimension*) of the category is shown on the left of the figure. These categories are similar to those presented in [9, 13], except that we have added direction information (horizontal and vertical). Finally, we note that each node in the lattice illustrates only one example of the corresponding gesture form. Other examples can be obtained by reflection across either axis, or by considering obtuse angles as opposed to acute angles (in the lower 2 levels of the lattice).

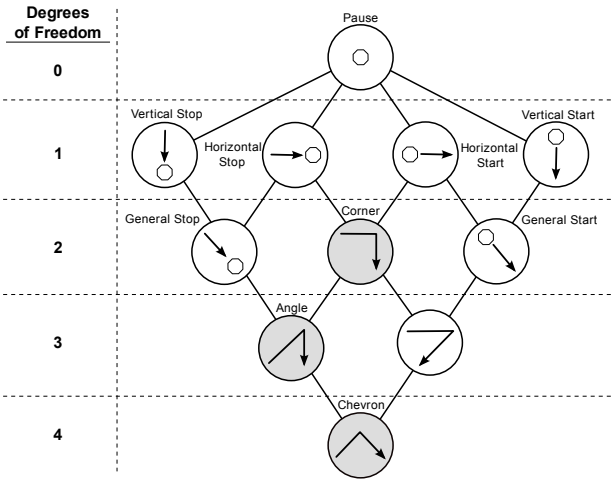

Figure 3: A hierarchy of gesture forms, each consisting of two motion segments. Each segment is described as either: horizontal movement, vertical movement, unconstrained movement or rest. Shaded regions correspond to the three gestures considered throughout this paper.

All forms in the hierarchy, except "chevron", have fewer than 4 degrees of freedom. Consequently, these forms have the potential to introduce non-accidental features. In this paper we consider gestures based only on the "corner", "angle", and "chevron" forms. In the "corner" gesture, the initial segment involves only movement in the positive x direction. Hence, this segment is parallel to the x-axis. The second segment involves only movement in the negative y direction, and is parallel to the y-axis. The resulting motion is highly specific, and allows for only two degrees of freedom. The "angle" gesture (so named on account of its similarity to the mathematical angle symbol $\angle$), has 3 degrees of freedom. The first segment of "angle" can generally be described as "up, and to the right", while the second segment remains "straight down". The "chevron" gesture has a full 4 degrees of freedom, and is not expected to yield any non-accidental features. This gesture can be described as first moving "up and to the right", and then moving "down and to the right". Table 1 lists the three gestures and their degrees of freedom.

	Corner	Angle	Chevron
X'_-	> 0	> 0	> 0
Y'_-	$= 0$	> 0	> 0
X'_+	$= 0$	$= 0$	> 0
Y'_+	< 0	< 0	< 0
d.o.f	2	3	4

Table 1: The three gestures used throughout this paper. Here X'_- and X'_+ denote the horizontal velocities of the first and second gesture segments respectively. Y'_- and Y'_+ are defined similarly.

Here we note that the most restricted gesture form, "pause", is often used in conjunction with gesture-based interfaces [6]. Perhaps most famously, pausing is the foundation of the "dwell click" gesture, where users make selections by pausing their hand (or stylus, or gaze) directly over a particular screen target. We opted against making explicit use of this feature because it can lead to the "Midas touch" problem [6], where gestures may inadvertently be activated whenever, and wherever, the hands rest.

4 Models supporting the discovery of regularities

In this section we describe the motion and mixture models used to discover regularities in the observations.

4.1 The motion model

Hand motion is modelled as a piecewise linear trajectory of the hand's centroid. In other words, the trajectory is composed of numerous linear segments. Within each segment, the hand moves with constant velocity. Of course, true hand motion is not so simple; the hand must go through phases of acceleration throughout the course of performing the gestures. In practice, such phases are short-lived. Moreover, the linear segments are easily recovered using the method described by Mann *et al.* in [14]. This approach uses dynamic programming to recover a minimum-cost segmentation of the hand trajectory into piecewise polynomial segments. The total segmentation cost is the sum of squared

errors in the polynomial fits, plus a fixed cost λ for each segment. In our case, the polynomials are taken to be of first order, and the cost function can be expressed as follows

$$\text{Cost} = \sum_{n=1}^{N} \left[\sum_{i=t_{n-1}}^{t_n} ||\mathbf{X}(t) - \hat{\mathbf{X}}_n(t;\theta_n)||^2 + \lambda \right] \quad (1)$$

where N is the total number of segments in the model, $\mathbf{X}(t) = [X(t)\ Y(t)]^T$ is the observed hand position at time t, and $\hat{\mathbf{X}}_n(t;\theta_n)$ is the n^{th} polynomial segment with coefficients θ_n.

Having established a piecewise linear model of motion, we now describe the feature space. At any instant t, the hand's motion can be parameterized by measuring the modelled hand position $\hat{\mathbf{X}}(t;\theta_n) = [\hat{X}(t;\theta_n)\ \ \hat{Y}(t;\theta_n)]^T$ and velocity $\hat{\mathbf{V}}(t;\theta_n) = [\hat{X}'(t;\theta_n)\ \ \hat{Y}'(t;\theta_n)]^T$. To simplify notation, we drop the parameterization by t and θ_n, giving $\hat{\mathbf{X}} = [\hat{X}\ \ \hat{Y}]^T$ and $\hat{\mathbf{V}} = [\hat{X}'\ \ \hat{Y}']^T$.

As a result of assuming zero acceleration, changes in velocity occur instantaneously. This introduces discontinuities in $\hat{X}'$ and $\hat{Y}'$. To account for the possibility of a discontinuity, the parameterization is augmented to include measurements of both the left and right partial derivatives, $\hat{\mathbf{V}}_- = [\hat{X}'_-\ \ \hat{Y}'_-]^T$ and $\hat{\mathbf{V}}_+ = [\hat{X}'_+\ \ \hat{Y}'_+]^T$ respectively. The motion at an instant is completely parameterized by the vector

$$[\hat{X}\ \ \hat{Y}\ \ \hat{X}'_-\ \ \hat{Y}'_-\ \ \hat{X}'_+\ \ \hat{Y}'_+]^T \quad (2)$$

4.2 Features

The gestures described in section 3 each include a single direction change. These direction changes are encoded as breakpoints in the motion model. Here breakpoints are instants where the motion model switches from one linear segment to the next. We select as features the motion parameterizations corresponding to the breakpoints. Since the gestures in this paper are translation invariant, the spatial parameters $\hat{X}$ and $\hat{Y}$ have little meaning and are ignored. The set of all possible features, $\mathbf{F}$, is the set of motion parameterizations where the pair of left-partial derivatives $[\hat{X}'_-\ \ \hat{Y}'_-]^T$ differs from the pair of right-partial derivatives $[\hat{X}'_+\ \ \hat{Y}'_+]^T$. This can be expressed as follows:

$$\mathbf{F} = \{\ [\hat{X}'_-\ \ \hat{Y}'_-\ \ \hat{X}'_+\ \ \hat{Y}'_+]^T\ \mid\ \hat{X}'_- \neq \hat{X}'_+\ \ \text{or}\ \ \hat{Y}'_- \neq \hat{Y}'_+\} \quad (3)$$

4.3 Mixture modelling

Performing gestures gives rise to various features according to the distribution $P(f|G)$ where $f \in \mathbf{F}$ is a specific feature and G is the gesture performed. Since the gestures involve exactly one direction change, at least one feature is expected for each performance. Additional features may also be introduced by more general changes in velocity. Our hypothesis is that some of these features are non-accidental. Non-accidental features imply that there are regularities associated with gesture, and these regularities are reflected by compact modes in $P(f|G)$.

The true distribution $P(f|G)$ is unknown, and must be modelled. A Gaussian mixture model is used for this purpose [15]. Each gesture G is modelled by a separate mixture model M_G, which generates features f according to the following:

$$P(f|M_G) = \sum_{i=k}^{K} \pi_k\ N(f\ ;\ \mu_k, \Sigma_k) \quad (4)$$

where π_k, $\sum_{k=1}^{K} \pi_k = 1$, is the prior probability of generating data from component k, and $N(f\ ;\ \mu_k, \Sigma_k)$ is the Gaussian distribution with mean μ_k and covariance matrix Σ_k for component k. To learn the model parameters from training data, we use the expectation maximization (EM) algorithm [15].

5 Implementation

Our experiments use the hand detection and tracking system originally developed for Maestro. 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. In our experiments, only the motion of the red glove was considered. The actual Maestro system tracks both gloves independently, allowing for the use of bimanual gestures.

6 Results

In order to perform the modelling described in section 4, a considerable amount of training data was required. Approximately 100 training examples were captured for each of the three gestures being modelled. Of these 100 training examples, 80 were used for learning the Gaussian mixture models, and the remaining 20 examples were withheld for validation purposes. Additionally, we captured several long sequences consisting of background motion with intermittent gestures.

Importantly, all gestures were performed in front of a white screen onto which a small target was projected. When performing each gesture, the hand motion was adjusted so that the required direction changes coincided with the hand reaching the target. After each performance, the target was randomly repositioned. Performing gestures at differing screen locations avoids the possibility of users repeating identical trajectories.

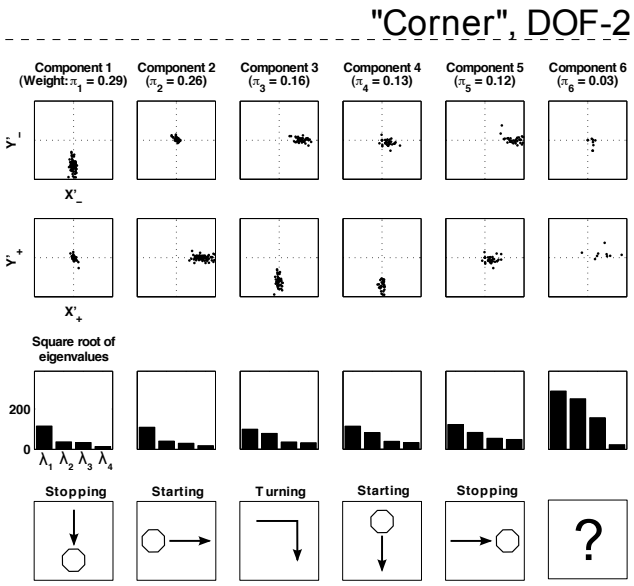

Figure 4: Component distributions learned for the "corner" gesture. Each column represents a different component. The first row corresponds to the velocity $\hat{\mathbf{V}}_-$ before the breakpoint. The second row corresponds to the velocity $\hat{\mathbf{V}}_+$ after the breakpoint. The third row displays the square root of the eigenvalues for each of the distribution covariance matrices Σ_i. The fourth row depicts our interpretation of the motion.

6.1 Clusters and potential regularities

Before learning the parameters of the mixture models, feature extraction was performed individually on each of the training examples (as described in section 4). On average, each training example contributed 4 features. As will be demonstrated below, these extra features arise from additional unexpected regularities associated with each gesture.

For each gesture class, all features were collected and these collections were then used to learn the mixture models. In all cases, the mixture models consisted of $K = 6$ mixture components. In the case of the "corner" gesture, the mixture components are visualized in figure 4.

Qualitatively, the clusters for the "corner" gesture look quite promising. In particular, we expected to see the component distribution depicted in figure 4, column 3. This distribution represents horizontal motion in the positive x direction, followed by vertical motion in the negative y direction. Somewhat unexpectedly, extra regularities occur in clusters 4 and 5. These clusters correspond to cases where there was a momentary pause between the horizontal and vertical segments. In this vein, cluster 1 represents the act of stopping upon completing the gesture, and cluster 2 represents the act of starting the gesture from rest. The final cluster is difficult to interpret. In any case, this cluster has low weight and does not contribute much to the overall mixture model.

Importantly, no single training example included features from all of the first 5 clusters. This suggests that there is some uncertainty in the *classes* of features that arise from the performance of any gesture. When "spotting" gestures, any approach that relies on the detection of a single regularity may miss some instances of the gesture.

Principal component analysis (PCA) provides a convenient means for analyzing the structure of the component distributions. With PCA, the covariance matrix Σ_i of the i^{th} component distribution undergoes an eigenvalue decomposition. The eigenvalues correspond to the variances along each of the principal axes. The standard deviations are recovered by taking the square root. These values are listed in the third row of figure 4.

Since covariance matrices are positive semi-definite, their eigenvalues are all non-negative. If any of the eigenvalues are zero, then the covariance matrix is singular, and the cluster's points lie in a lower dimensional subspace of the feature space. Provided that the corresponding mixture component has positive weight, the cluster corresponds to a regularity, and its features are considered to be non-accidental.

Unfortunately, singular covariance matrices are almost never observed in real data. This is because each feature incorporates some level of noise. Rather than searching for singular covariance matrices, we search for heavily weighted clusters with covariance matrices containing exceptionally low eigenvalues. From figure 4, it is clear that the first 5 components of the "corner" gesture fit this description. These components are potential regularities. Moreover, the first two components each have three low eigenvalues. This suggests that the "starting" and "stopping" components have fewer degrees of freedom as compared to the "turning" component which has only two low eigenvalues. This corresponds well with the starting and stopping forms depicted in the lattice described in figure 3.

The component distributions corresponding to the "angle" and "chevron" gestures are listed in figures 5 and 6. Each of these gestures contributes three potential regularities. Qualitatively, their component distributions appear somewhat less compact than those observed for the "corner" gesture. This trend is also apparent in general scattergrams of the features before clustering (figure 7).

6.2 Validation

We now turn to the problem of spotting gestures in sequences containing intermittent background motion. Since the gestures were motivated by non-accidental features, it is natural to assume gestures occur in their vicinity. This leads to a remarkably simple gesture spotting approach where

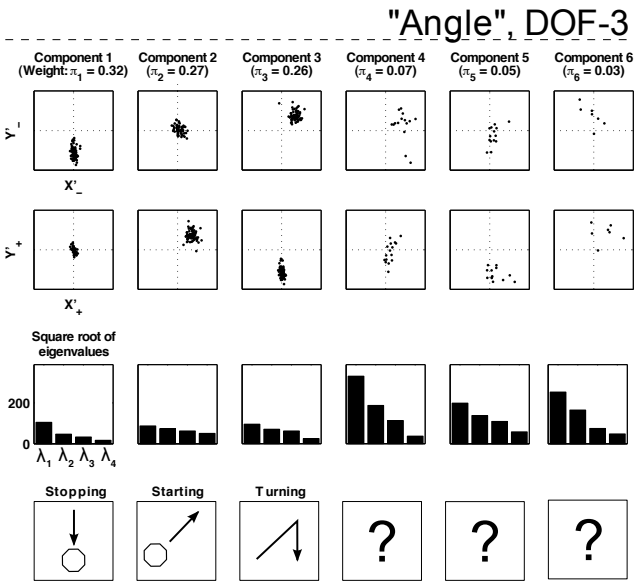

Figure 5: Component distributions learned for the "angle" gesture (compare with figure 4).

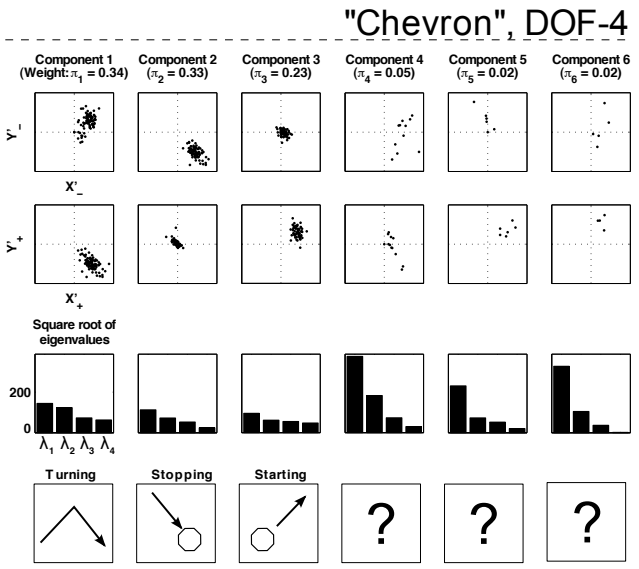

Figure 6: Component distributions learned for the "chevron" gesture (compare with figure 4).

every feature is classified as either belonging to the background or as being non-accidental (and arising from a gesture). Classification makes use of the entire gesture mixture model, and thus indirectly compares each observed feature to all regularities associated with the gesture. If *any* features are positively classified, an *entire* gesture instance is considered spotted. Contiguous positive classifications are considered to arise from a single gesture instance.

Ideally, to classify a feature as arising from either gesture or background motion, one requires access to a model for expected background motion. If such a model were known,

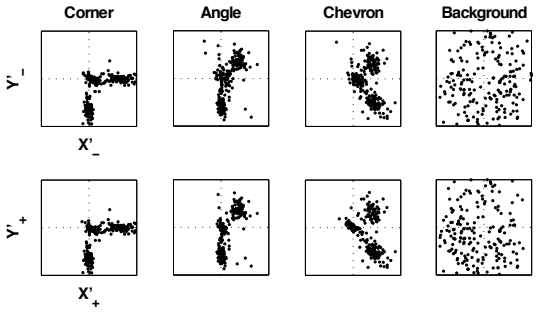

Figure 7: Features observed for each of the three gestures, as well as for a sample of unconstrained background motion.

then classification could proceed using a maximum likelihood approach. Unfortunately, we do not have access to such a model. Instead the feature f is considered to arise from a gesture if $P(f|M_G) > \alpha$, where α is a constant. For this approach to be successful, known gesture features f_G should be assigned much higher likelihood by the model as compared to known background features f_B. This can be captured by the following inequality:

$$log\left[\frac{1}{N}\sum_{i=1}^{N}P(f_{G,i}|M_G)\right] > log\left[\frac{1}{M}\sum_{i=1}^{M}P(f_{B,i}|M_G)\right] \quad (5)$$

The expression on either side of the inequality can be interpreted as the "log of the average likelihood". The terms $f_{G,i}$ and $f_{B,i}$ represent the i^{th} features attributed to the gesture or to the background respectively. There are N features known to have been generated from the gesture and M features known to have been generated by the background. In order to check that this condition is satisfied, each gesture model was tested against the held out training data as well as to recordings of background hand motion. The results are presented in figure 8.

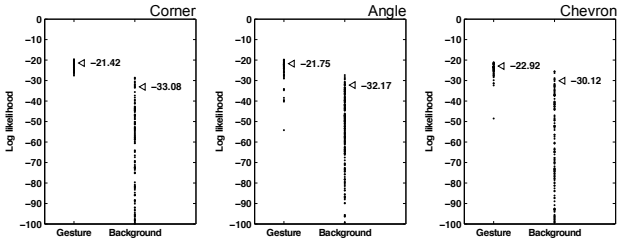

Figure 8: (Each scattergram) Points depict the log likelihood $logP(f|M_G)$ of either a known gesture feature (left), or a known background feature (right). The arrows indicate the average of the corresponding likelihoods.

For all three gestures, the inequality holds. Consequently, we expect the aforementioned classification pro-

cedure to be reasonably effective at classifying features. In order to quantify this effectiveness we apply the classification procedure to motion sequences in which both gestures and background motion are present. The timing of the gestures within these sequences is known in advance. This establishes a ground-truth labelling of the features, and allows for the measurement of the classifier's *precision* and *recall*. *Precision* is the percentage of correct classifications amongst all features classified as arising from gestures. *Recall* is the percentage of *all* gesture features that were classified correctly. These measures can be expressed mathematically as follows:

$$\text{precision} = \frac{\text{\# of true positives}}{\text{\# of true positives} + \text{\# of false positives}} \quad (6)$$

$$\text{recall} = \frac{\text{\# of true positives}}{\text{\# of true positives} + \text{\# of false negatives}} \quad (7)$$

With the classification procedure described above, both the precision and the recall scores depend directly on the likelihood threshold α; precision increases with α, while recall decreases. Since our work is motivated by the Maestro presentation system, achieving a high precision is of the utmost importance. When giving a presentation, presenters are far more tolerant of false negatives than they are to false positives [5]. In order to ensure a high level of precision we set α to the lowest value possible while maintaining a precision of ≥ 0.95. The recall achieved at this level of precision is used as one measure of the effectiveness of the models at spotting gestures. The results of these experiments are listed in table 2.

d.o.f	Gesture	Recall	Recall*
2	"corner"	0.84	1.00
3	"angle"	0.67	1.00
4	"chevron"	0.40	0.79

Table 2: Recall scores for each of the 3 gestures, where precision is fixed to ≈ 0.95

While the recall score is informative, it is perhaps not an entirely fair measure of the effectiveness of "spotting" gestures. In order to "spot" a gesture, only one of its many features needs to be recognized as arising from the gesture. We therefore redefine recall to be the percentage of gestures which result in the positive classification of *at least one* feature. We refer to this measure as recall*, and report the corresponding scores in the fourth column of table 2.

For both the recall and the recall* measures, the "corner" classifier scores better than the "angle" classifier, which in turn scores better than the "chevron" classifier. This difference is depicted graphically in figure 9, which plots the log likelihood of each model over time. Notice that as the degrees of freedom increase, so does the threshold α. This corresponds to fewer gesture features falling above this threshold, resulting in the drop of recall performance that was reported in table 2. While preliminary, the results suggest that classification performance becomes more reliable as dimension decreases. Furthermore, we may consider "chevron" to be an arbitrary gesture with a full 4 degrees of freedom. The "chevron" results provide a baseline performance for when non-accidental features are not used.

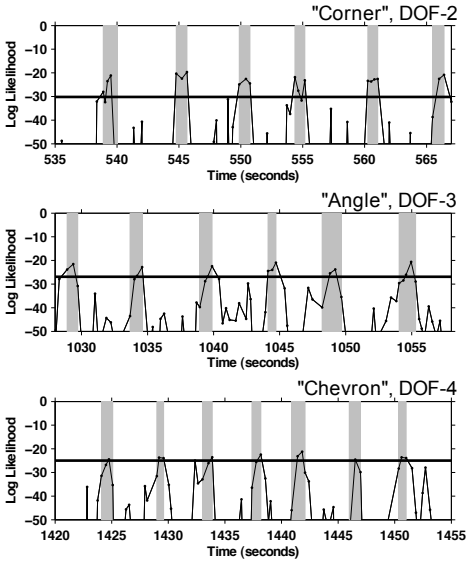

Figure 9: Log likelihood $P(f|M_G)$ of the model over time (as new features arrive). Shaded regions indicate intervals in which the gesture is known to have occurred. The thick horizontal line indicates the likelihood threshold α. Each timeline shows 6 of the 20 gesture instances used to test each category.

The aforementioned results are for the complete gesture mixture models. In section 6.1 it was argued that only a few components of each model are potential regularities to which non-accidental features are attributed. Consequently, it is expected that successful gesture spotting is driven primarily by these components. To examine if this is indeed the case, figure 10 breaks down the log likelihood of the "corner" model into each of its component distributions. Here the same data was used as in figure 9. The "corner" gesture gave rise to 5 potential regularities, and in figure 10 these components exhibit strong responses to the gesture.

7 Discussion

In this paper we have shown that non-accidental features do occur when people perform certain gestures, and that these features can be reliably detected. This was demonstrated by learning Gaussian mixture models of the features associated with gesture. Non-accidental features resulted in compact, heavily-weighted, mixture component distribu-

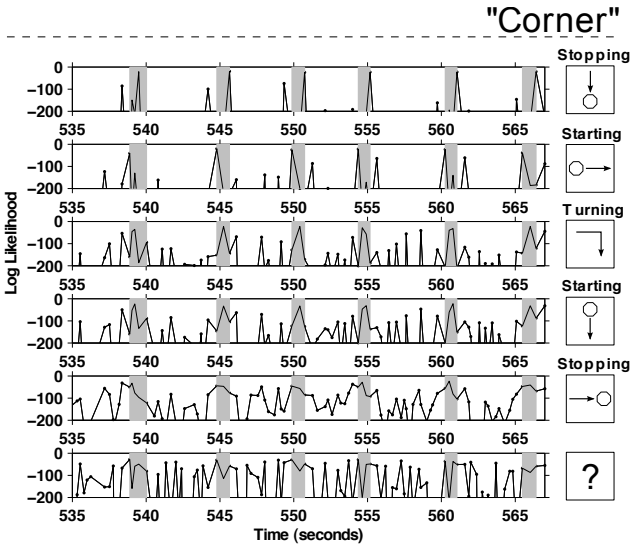

Figure 10: Log likelihood vs. time for each component distribution of the "corner" gesture model. To the right of each time line lies a pictorial representation of the expected motion.

tions. Reliable detection was demonstrated by illustrating how the mixture models provide effective means for discriminating non-accidental features from the background.

In addition to the aforementioned primary results, an unexpected result was the discovery of highly regular gesture fragments (such as stopping and then restarting midgesture). These fragments were not included in the original gesture specifications, but emerged as regularities nonetheless. This may indicate that people hesitate briefly, between the strokes of the gesture. Possible future work would be to exploit the temporal constraints within the segmented trajectories. Additionally, we will consider non-accidental features associated with two-handed motion, and the spatial relations between the hands and items on the screen. In fact, cognitive scientists have already discovered that spatial relations are used by subjects to describe object animacy [17].

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

[3] 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.

[4] J. Feldman. Constructing perceptual categories. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 244–250, Jun 1992.

[5] A. Fourney, M. Terry, and R. Mann. The presentation maestro: Direct manipulation through gesture alone. Technical report, David R. Cheriton School of Computer Science, University of Waterloo, 2009.

[6] 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.

[7] A. Jepson and R. Mann. Qualitative probabilities for image interpretation. In *ICCV99*, pages 1123–1130, 1999.

[8] 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).

[9] A. D. Jepson and J. Feldman. A biased view of perceivers. In D. Knill and W. Richards, editors, *Perception as Bayesian Inference*, pages 229–235. Cambridge University Press, 1996.

[10] D. Kim, J. Song, and D. Kim. Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms. *Pattern Recogn.*, 40(11):3012–3026, 2007.

[11] 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.

[12] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Norwell, MA, USA, 1985.

[13] 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.

[14] R. Mann, A. D. Jepson, and T. El-Maraghi. Trajectory segmentation using dynamic programming. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, page 10331, Washington, DC, USA, 2002. IEEE Computer Society.

[15] G. J. McLachlan and K. E. Basford. *Mixture models. Inference and applications to clustering*. Statistics: Textbooks and Monographs, New York: Dekker, 1988.

[16] D. Tausky and R. Mann. Categorization and learning of pen motion using hidden markov models. *Computer and Robot Vision, Canadian Conference*, 0:488–495, 2004.

[17] P. D. Tremoulet and F. J. Perception of animacy from the motion of a single object. *Perception*, 29:943–951, 2000.

[18] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[19] 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.

[20] 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.