M.I.T. Media Laboratory Perceptual Computing Section Technical Report No. 421
Submitted to: International Conference on Computer Vision, 1998

Recognition and Interpretation of Parametric Gesture

Andrew D. Wilson Aaron F. Bobick
Vision and Modeling Group
MIT Media Laboratory
20 Ames St., Cambridge, MA 02139
(drew, bobick@media.mit.edu)

Abstract

A new method for the representation, recognition, and interpretation of parameterized gesture is presented. By parameterized gesture we mean gestures that exhibit a meaningful variation; one example is a point gesture where the important parameter is direction. Our approach is to extend the standard hidden Markov model method of gesture recognition by including a global parametric variation in the output probabilities of the states of the HMM. Using a linear model to derive the theory, we formulate an expectation-maximization (EM) method for training the parametric HMM. During testing, the parametric HMM simultaneously recognizes the gesture and estimates the quantifying parameters. Using visually-derived and directly measured 3-dimensional hand position measurements as input, we present results on two different movements — a size gesture and a point gesture — and show robustness with respect to noise in the input features.

1 Introduction

Current approaches to the recognition of human movement work by matching an incoming signal to a set of representations of prototype sequences. For example, a gesture recognition system might match a sequence of hand positions over time to a number of prototype gesture sequences, each of which are learned from some number of examples. To handle variations in temporal behavior, the match is typically computed using some form of dynamic time warping (DTW). If the prototype is described by statistical tendencies, the time warping is often embedded within a hidden Markov model (HMM) framework. When the match to a particular prototype is above some threshold, the system concludes that the gesture corresponding to that prototype has occurred.

Consider, however, the problem of recognizing the gesture pictured in Figure 1 that accompanies the speech "I caught a fish. It was *this* big." The gesture co-occurs with the word "this" and is intended to convey a quantity, namely the size of the fish. The difficulty in recognizing this gesture is that its form varies greatly depending on this quantity. A simple DTW or HMM approach would attempt to model this important relationship as noise. We call movements that exhibit meaningful, systematic variation *parameterized movements*.

Many hand gestures that accompany speech may be considered parameterized movements. As with the "fish" example, hand gestures are often used in dialog to convey some quantity that otherwise cannot be determined from the speech alone. Parameterized movements are also extensively used in musical conducting,

Figure 1: The gesture that accompanies the speech "I caught a fish. It was *this* big." In its entirety, the gesture consists of a preparation phase in which the hands are brought into the gesture space, a stroke phase (depicted by the illustration) which co-occurs with the word "this" and finally a retraction back to the rest-state (hands down and relaxed). The distance θ conveys the size of the fish.

where the quality of the conductor's movement is as important as the presence of the movement itself. Driving is another domain rich with meaningful parameterized movements.

Techniques that use fixed prototypes for matching are not well suited to modeling movements that exhibit such meaningful variation. In this paper we present a framework which models parameterized movements in a such way that the recovery of the parameter of interest and the recognition of the movement proceed simultaneously.

In this paper we extend the standard hidden Markov model method of gesture recognition to include a global parametric variation in the output probabilities of the states of the HMM. Using a linear model to derive the theory, we formulate an expectation-maximization (EM) method for training the parametric HMM. During testing, the parametric HMM simultaneously recognizes the gesture and estimates the quantifying parameters. Using visually-derived and directly measured 3-dimensional hand position measurements as input, we present results on two different movements — a size gesture and a point gesture — and show robustness with respect to noise in the input features.

2 Related work

Hidden Markov models and related techniques have been applied to the problem of gesture recognition with notable success. For example, Darrell and Pentland [8] applied dynamic time warping to match image template correlation scores against models to recognize hand gestures from video. A gesture model is represented by a temporal pattern of correlation scores. Schlenzig, Hunter and Jain [17] used HMMs and a rotation-invariant image representation to recognize hand gestures from video. Starner and Pentland [18] applied HMMs to recognize ASL sentences.

None of these works have developed representations to learn meaningful variation of the gestures. For example, Starner and Pentland restrict the ASL alphabet to repeatable, non-varying gestures. In fact ASL is subject to complex grammatical processes that operate on multiple simultaneous levels. These kinds of variation in ASL are addressed in a machine perception framework by Poizner et al [15].

A number of systems have been developed which use gesture recognition within an interactive context. These are relevant for the present work in that the system is charged with the task of extracting a parameter important to the interaction as well as the task of recognizing that the gesture occurred. The ALIVE [9] and Perseus [12] systems are examples. The typical approach of these systems is to first identify static configurations of the user's body that are diagnostic of the gesture, and then use an unrelated method to extract the parameter of interest (e.g., direction of pointing). Manually constructed ad hoc procedures are typically used to identify the diagnostic configuration, a task complicated by the requirement that this procedure work through the range of meaningful variation and also not be confused by other gestures. Perseus, for example, understands pointing gestures by detecting when the user's arm is extended. The system then finds the pointing direction by computing the line from the head to the user's hand.

Darrell [7] addresses the problem of crafting perceptual strategies automatically in part by training a model of attention. The model learns a policy to select features from the input using partially observable Markov decision process (POMDP). In [1] we used hand-tuned HMMs using temporal properties to recognize two broad classes of natural, spontaneous gesture. Campbell and Bobick [6] search for orthogonal projections of the feature space to find the most diagnostic projections in order to classify ballet steps.

In [20], we apply HMMs to the task of hand gesture recognition from video by training an eigenvector basis set of the images at each state. An image's membership to each state is a function of the residual of the reconstruction of the image using the state's eigenvectors. The state membership is thus invariant to variance along the eigenvectors. Although not applied to images directly, the present work is an extension of this earlier work in that the goal is to recover a parameterization of the systematic variation of the gesture.

Murase and Nayar [13] parameterize meaningful variation in the appearance of images by computing a representation of the nonlinear manifold of the images in an eigenspace of the images. Their work is similar to ours in that training assumes that each input feature vector is labeled with the value of the parameterization. In testing, an unknown image is projected onto the manifold and the parameterization is recovered. Their framework has been used, for example, to recover the camera angle relative to a known object in the field of view.

Parameterized object recognition has sought to couple the matching problem with the estimation of parameters of the object. For example, ACRONYM (Brooks [4]) uses constraints on the free parameters that specify the shape of an object. The recognition is then a matter of satisfying these constraints and setting the free parameters. Potential approaches to parameterized object recognition are discussed in [10].

Recently there has been interest in methods that recover latent parameterizations. In his "family discovery" paradigm, Omohundro [14], for example, outlines a variety of approaches to learning the nonlinear manifold representing systematic variation. One of these techniques has been applied to the task of lip reading by Bregler and Omohundro. Bishop, Svensen and Williams [5] have also introduced techniques to learn latent parameterizations. Their system begins with an assumption of the dimensionality of the parameterization and uses an expectation-maximization framework to compute a manifold.

Lastly we mention Tenenbaum and Freeman's work on separating style from content. They use a generative factorial model to learn separate representations of style (e.g., font) and content (e.g., letter). Their work differs from ours in that the emphasis is learning the factorial structure of the problem. However, the goal of learning variation due to style is common to both.

2.1 Images versus features

Previous work in gesture recognition, including some of our own, has been criticized for using features that are not robust with respect to varying imaging conditions (e.g. illumination changes or translation of the hand's rest position with respect to the camera). In this paper we describe the results of two experiments. The first test operates on hand position features derived from a stereo vision system; the second, position features generated using a electromagnetic sensor. Our view is that the extraction of robust, salient, semantically rich features is an important problem but one that is independent of the recognition and estimation task being investigated here. Our technique however does not require noiseless feature tracking: in the results section we show the performance of our technique under varying noise conditions.

2.2 Non-parametric extensions

Before presenting our method for modeling parameterized movements, it is worthwhile to consider two extensions of the standard gesture recognition paradigm to the problem of recognizing these parameterized classes.

The first approach relies on our ability to come up with *ad hoc* methods to extract the value of the parameter of interest. For example, in the example presented in Figure 1, one could design a procedure to recover the parameter: wait until the hands are in the middle of the gesture space and have low velocity, then calculate the distance between the hands. One example of such approach is contained in the Perseus system ([12]).

The chief objection to such an approach is not that each movement requires a new *ad hoc* procedure, nor the difficulty in writing procedures that recover the parameter *robustly*, but the fact that they are only appropriate to use when the gesture has already been labeled. As mentioned in the introduction, a recognition system that abstracts over the variation induced by the parameterization must model such variation as noise or deviation from a prototype. The greater the parametric variation, the less constrained the recognition prototype can be, and the worse the detection results become.

The second approach employs multiple DTW or HMM models to cover the parameter space. Each DTW model or HMM is associated with a point in parameter space. In learning, the problem of

allocating training examples labeled by a continuous variable to one of a discrete set of models is eliminated by uniting the models in a mixture of experts framework [11]. In testing, the parameter is extracted by finding the best match among the models and looking up its associated parameter value. The dependency of the movement's form on the parameter is thus removed. This can be embellished somewhat by computing the value of the parameter as the weighted average of all the models' associated parameter values, where the weights are derived from the matching process.

The first objection to this approach is that it is unknown from the outset how many separate models will be necessary. The second objection is that all of the models are required to learn the same or similar dynamics (i.e. as modeled by the transition matrix in the case of HMMs) separately. The last objection is the most serious in terms of practical use: as the dimensionality of the parameter space increases, the large number of models necessary to cover the space will place unreasonable demands on the amount of training data. [1]

In the next section we introduce parametric HMMs, which overcome the problems with both approaches presented above.

3 Parametric hidden Markov models

3.1 Model

Parametric HMMs model the dependence on the parameter of interest explicitly. We begin with the usual HMM formulation [16] and change the form of the output probability distribution (usually a normal distribution or a mixture model) to depend on the parameter θ, a vector quantity.

In the standard continuous HMM model, a sequence is represented by movement through a set of hidden states. The Markovian property is encoded in a set of transition probabilities, with $a_{ij} = P(q_t = j \mid q_{t-1} = i)$ being the probability of moving to state j at time t given the system was in state i at time $t-1$. Associated with each state j is an output distribution of the feature vector $\mathbf{x}$ given the system is really in state j at time t: $P(\mathbf{x}_t \mid q_t = j)$. In a simple Gaussian HMM, the parameters to be estimated are the a_{ij}, μ_j, and Σ_j.[2]

To introduce the parameterization on θ we modify the output distributions. The simplest useful model is a linear dependence of the mean of the Gaussian on θ. For each state j of the HMM we have:

$$\hat{\mu}_j(\theta) = W_j\theta + \bar{\mu}_j \tag{1}$$

$$P(\mathbf{x}_t \mid q_t = j, \theta) = \mathcal{N}(\mathbf{x}_t, \hat{\mu}_j(\theta), \Sigma_j) \tag{2}$$

In the work presented here all values of θ are considered equally likely and so the prior $P(\theta \mid q_t = j)$ is ignored.

Note that θ is constant for the entire observation sequence, but is free to vary from sequence to sequence. When necessary, we write the value of θ associated with a particular sequence k as θ_k.

[1] In such a situation it is not sufficient to simply interpolate the match scores of just a few models in a high dimensional space since either (1) there will be significant portions of the space for which there is no response from any model or (2) in a mixture of experts framework, each model is called on to model too much of the space, and so is modeling the dependency on the parameter as noise.

[2] Technically there are also the initial state parameters π_j to be estimated; in this work we use causal topologies with a unique starting state.

3.2 Training

Training consists of setting the HMM parameters to maximize the probability of the training sequences. Each training sequence is paired with a value of theta. The Baum-Welch form of the expectation-maximization (EM) algorithm is used to update the parameters of the output probability distributions. The expectation step of the Baum-Welch algorithm (also known as the "forward/backward" algorithm) computes the probability that the HMM was in state j at time t given the entire sequence $\mathbf{x}_t$ denoted as γ_{tj}. It is convenient to consider the HMM's parse of the observation sequence as being represented by γ_{tj}.

In training, the parameters ϕ of the HMM are updated in the maximization step of the EM algorithm. In particular, the parameters ϕ are updated by choosing a ϕ' to maximize the auxiliary function $Q(\phi' \mid \phi)$. ϕ' may contain all the parameters in ϕ, or only a subset if several maximization steps are required to estimate all the parameters. As explained in the appendix, Q is the expected value of the log probability given the parse γ_{tj}. In the appendix we derive the derivative of Q for HMM's:

$$\frac{\delta Q}{\delta \phi'} = \sum_t \sum_j \gamma_{tj} \frac{\frac{\delta}{\delta \phi'} P(\mathbf{x}_t \mid q_t = j, \phi')}{P(\mathbf{x}_t \mid q_t = j, \phi')} \tag{3}$$

The parameters ϕ of the parameterized Gaussian HMM include W_j, $\bar{\mu}_j$, Σ_j and the Markov model transition probabilities. Updating W_j and $\bar{\mu}_j$ separately has the drawback that when estimating W_j only the old value of $\bar{\mu}_j$ is available, and similarly if $\bar{\mu}_j$ is estimated first. Instead, we define new variables:

$$Z_j \equiv \begin{bmatrix} W_j & \bar{\mu}_j \end{bmatrix} \quad \Omega_k \equiv \begin{bmatrix} \theta_k \\ 1 \end{bmatrix} \tag{4}$$

such that $\hat{\mu}_j = Z_j\Omega_j$. We then need to only update Z_j in the maximization step for the means.

To derive an update equation for Z_j we maximize Q by setting equation 3 to zero (selecting Z_j as the parameters in ϕ') and solving for Z_j. Note that because each observation sequence k in the training set is associated with a particular θ_k, we can consider all observation sequences in the training set before updating Z_j. Accordingly we denote γ_{tj} associated with sequence k as γ_{ktj}. Substituting the Gaussian distribution and the definition of $\hat{\mu}_j = Z_j\Omega_j$ into equation 3:

$$\frac{\delta Q}{\delta Z_j} = \sum_k \sum_t \gamma_{ktj} \Sigma_j^{-1} (\mathbf{x}_{kt} - \hat{\mu}_j(\theta_k)) \frac{\delta \hat{\mu}_j(\theta_k)}{\delta Z_j} \tag{5}$$

$$= \Sigma_j^{-1} \sum_k \sum_t \gamma_{ktj} (\mathbf{x}_{kt} - \hat{\mu}_j(\theta_k)) \Omega_k^T \tag{6}$$

$$= \Sigma_j^{-1} \left[\sum_{k,t} \gamma_{ktj} \mathbf{x}_{kt} \Omega_k^T - \sum_{k,t} \gamma_{ktj} Z_j \Omega_k \Omega_k^T \right] \tag{7}$$

Setting this derivative to zero and solving for Z_j, we get the update equation for Z_j:

$$Z_j = \left[\sum_{k,t} \gamma_{ktj} \mathbf{x}_{kt} \Omega_k^T \right] \left[\sum_{k,t} \gamma_{ktj} \Omega_k \Omega_k^T \right]^{-1} \tag{8}$$

Once the means are estimated, the covariance matrices Σ_j are updated in the usual way:

$$\Sigma_j = \sum_{k,t} \frac{\gamma_{ktj}}{\sum_t \gamma_{ktj}} (\mathbf{x}_{kt} - \hat{\mu}_j(\theta_k))(\mathbf{x}_{kt} - \hat{\mu}_j(\theta_k))^T \tag{9}$$

as is the matrix of transition probabilities [16].

3.3 Testing

In testing we are given an HMM and an input sequence. We wish to compute the value of θ and the probability that the HMM produced the sequence. As compared to the usual HMM formulation, the parameterized HMM's testing procedure is complicated by the dependence of the parse on the unknown θ. Here we present only a technique to extract the value of θ, since for a given value of θ the probability of the sequence $\mathbf{x}_t$ is easily computed by the Viterbi algorithm or by the forward/backward algorithm.

We desire the value of θ which maximizes the probability of the observation sequence. Again an EM algorithm is appropriate: the expectation step is the same forward/backward algorithm used in training. The forward/backward algorithm computes the optimal parse given a value of θ. In the corresponding maximization step we update θ to maximize Q, the log probability of the sequence given the parse γ_{tj}.

To derive an update equation for θ, we start with the derivative in equation 3 from the previous section and select θ as ϕ'. As with Z_j, only the means μ_j depend upon θ yielding:

$$\frac{\delta Q}{\delta \theta} = \sum_t \sum_j \gamma_{tj} \Sigma_j^{-1} (\mathbf{x}_i - \hat{\mu}_j(\theta)) \frac{\delta \hat{\mu}_j(\theta)}{\delta \theta} \qquad (10)$$

Setting this derivative to zero and solving for θ, we have:

$$\theta = \left[\sum_{t,j} \gamma_{tj} W_j^T \Sigma_j^{-1} W_j \right]^{-1} \left[\sum_{t,j} \gamma_{tj} W_j^T \Sigma_j^{-1} (\mathbf{x}_t - \bar{\mu}_j) \right] \qquad (11)$$

The values of γ_{tj} and θ are iteratively updated until the change in θ is small. With the examples we have tried, less than ten iterations are sufficient. Note that for efficiency, many of the inner terms of the above expression may be pre-computed.

4 Results

In our first testing result we make good on the example discussed in the introduction: "I caught a fish. It was *this* big." In the second test we show more detailed results for a pointing gesture, which is naturally parameterized by two values.

4.1 Size gesture

To test the ability of the parameterized HMM to learn the parameterization, thirty examples of the type depicted in Figure 1 were collected using the Stereo Interactive Virtual Environment (STIVE)[2], a research computer vision system utilizing wide baseline stereo cameras and flesh tracking (see Figure 2). STIVE is able to compute the three-dimensional position of the head and hands at a frame rate of about 20Hz.

The 30 sequences averaged about 43 samples in length. The actual value of θ, which in this case is interpreted as a size in inches, was measured directly by finding the point in each sequence during which the hands were stationary and then computing the distance between the hands. The value of θ varied from 7.7 inches (a small fish) to 36.6 inches (a respectable catch).

A six state parameterized HMM was trained with fifteen sequences randomly selected from the pool of thirty. The topology of the HMM was set to be causal (i.e., no transitions to previously visited states, with no "skip states"). In this example ten iterations were required for convergence, when the relative change in the total log probability for the training examples was less than one part in one thousand.

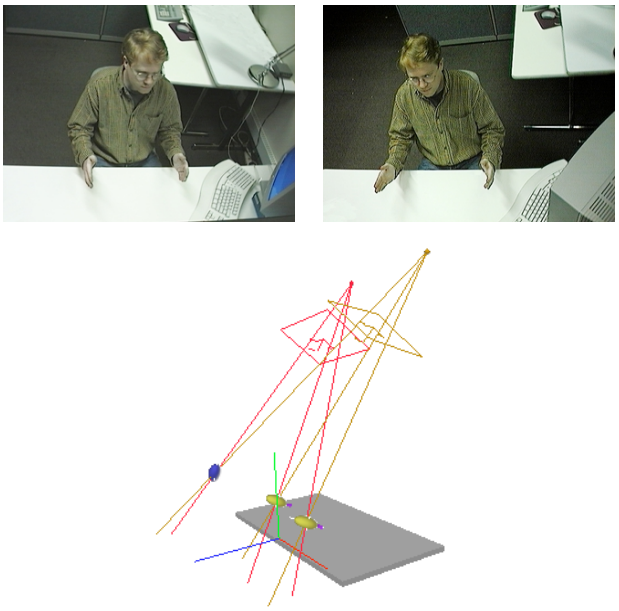

Figure 2: The Stereo Interactive Virtual Environment (STIVE) computer vision system used to collect data in section 4.1. Using flesh tracking techniques, STIVE computes the three-dimensional position of the head and hands at a frame rate of about 20Hz.

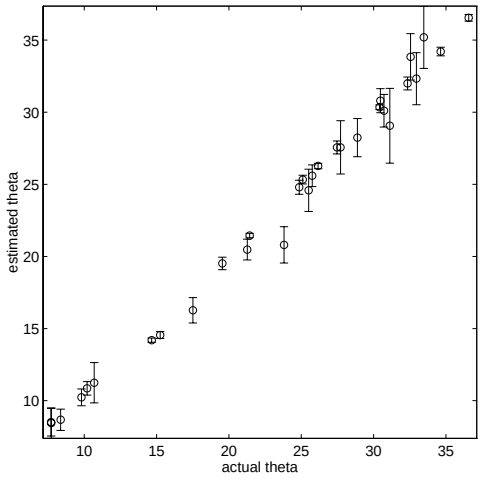

Figure 3: Parameter estimation results for the size gesture. Fifty random choices of the test and training sets were used to compute mean and standard deviation (error bars) on all examples. The HMM was retrained for each choice of test and training set.

Testing was performed with the remaining fifteen sequences. The size parameter θ was extracted from each of the testing sequences. To evaluate the performance of the parameterized HMM, we calculated the difference between the estimated value of θ and the value computed by direct measurement.

Figure 3 shows statistics on the parameter estimation for fifty (random) choices of the test and training sets. The HMM was

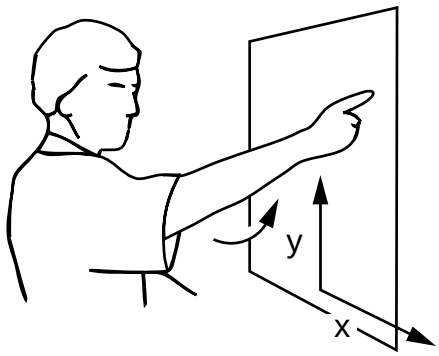

Figure 4: The point gesture used in section 4.2. The movement is parameterized by the coordinates of the target $\theta = (x, y)$ within a plane in front of the user. The gesture consists of a preparation phase, a stroke phase (shown here) and a retraction.

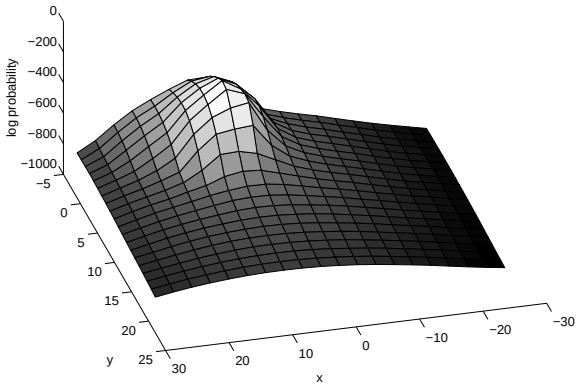

Figure 5: Log probability as a function of $\theta = (x, y)$ for a pointing test sequence.

retrained for each choice of test and training set. The average absolute error over all test trials is about 0.16 inches, demonstrating that the parameterized HMM had learned the parameterization accurately. Also, the magnitude of the W_j is greatest for the states corresponding to the middle phase of the gesture where the variation of θ maximally impacts the execution of the gesture. The system automatically learns which segment in the gesture that is most diagnostic of θ.

4.2 Pointing gesture

In the second test, we demonstrate the application of the parameterized HMM technique to a gesture parameterized by more than one variable. We also demonstrate the performance of the technique under varying amounts of noise and show the performance in a test requiring simultaneous recognition of the gesture and the extraction of the parameter θ.

For a movement that requires a multi-dimensional parameterization, we chose the pointing gesture. If pointing direction is restricted to the hemisphere in front of the user, the movement is naturally parameterized by a position in a plane in front of the user (see Figure 4). Neglecting the shape of the hand we recorded the three-dimensional position of the wrist (right hand). We used a

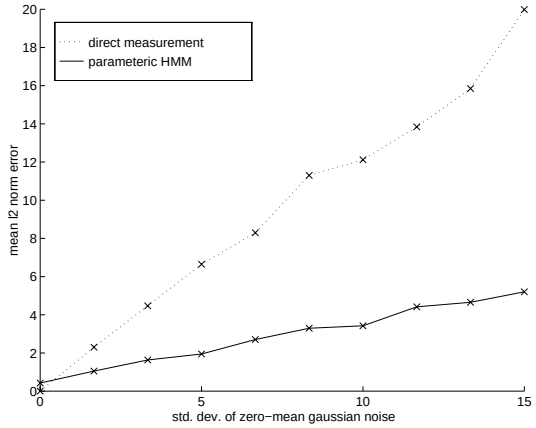

Figure 6: Average error over the entire pointing test set as a function of noise. The value of θ was estimated by an direct measurement and by a parameterized HMM retrained for each noise condition. The average error was computed by comparing the estimate of θ to the value recovered by direct measurement in the noise-free case.

Polhemus motion capture system to record the wrist position at a frame rate of 30Hz.

Fifty such examples were collected, each averaging 29 time samples (about 1 second) in length. As ground truth, we again directly measured the value of θ for each sequence: the point at which the depth of the wrist away from the user was found to be greatest. The position of this point in the pointing plane was returned. The horizontal coordinate of the pointing target varied from -22 to +27 inches, while the vertical coordinate varied from -4 to +31 inches.

An eight state causal parameterized HMM was trained using twenty sequences randomly selected from the pool of fifty. The remaining thirty sequences were used to test the ability of the model to encode the parameterization. The average error was computed to be about 0.37 inches (about 0.5 degrees). When the number of training examples was cut to 5 randomly selected sequences, the error increased to 0.82 inches (about 1.1 degrees), demonstrating how the parameterized HMM can exploit interpolation to reduce the amount of training data necessary. Again, the high level of accuracy can be explained by the increase in the weights W_j in those states that are most sensitive to variation in θ.

Parameterized HMMs examine the entire sequence to recover θ. For classes of movement in which there is systematic variation throughout the extent of the sequence, parameterized HMMs should perform more robustly than techniques that rely on querying a single point in time. To show this ability, we added various amounts of Gaussian noise to both the training and test sets, and then estimated θ using the direct measurement procedure outlined above and again with the parameterized HMM testing EM procedure. The parameterized HMM was retrained for each noise condition. For both cases the average error in parameter estimation was computed by comparing the estimated value with the value as measured directly with no noise present. The average error, shown in Figure 6, indicates that the parameterized HMM is more robust to noise than the *ad hoc* technique.

One concern in the use of EM for optimization is that while each EM iteration will increase the probability of the observations, there is no guarantee that EM will find the global maximum of

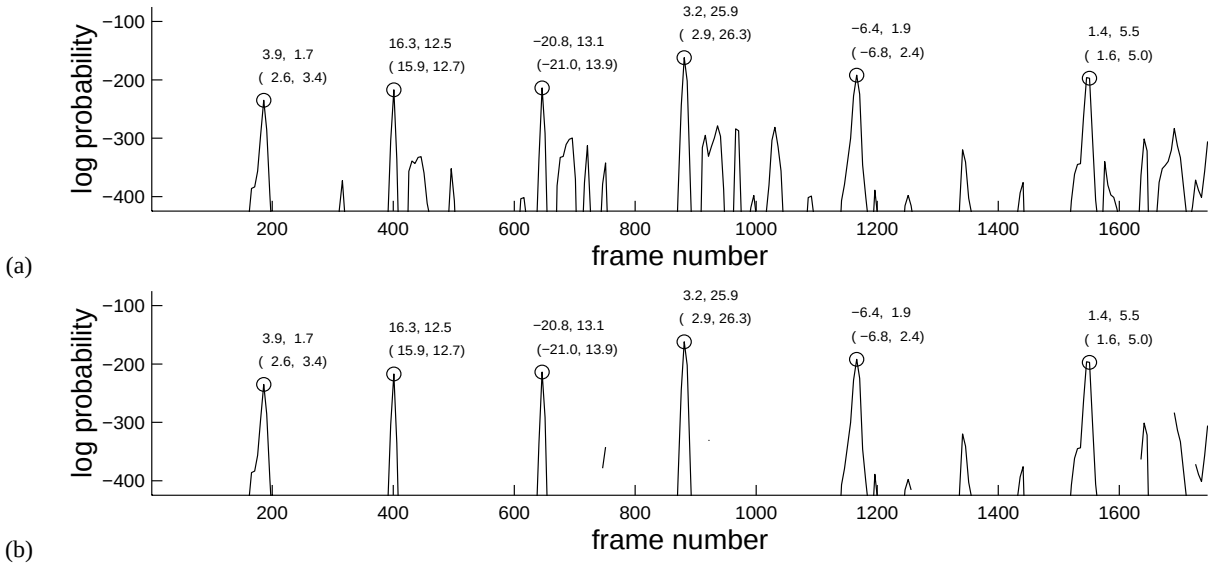

Figure 7: Recognition results are shown by the log probability of the windowed sequence beginning at each frame number. The true positive sequences are labeled by the value of θ recovered by the EM testing algorithm and the error computed by comparing the extracted value with the actual value computed by direct measurement. (a) Maximum likelihood estimate. (b) Maximum a posterior estimate, for which a uniform prior probability on θ was determined by the bounds of the training set. The MAP estimate was computed by simply throwing out sequences for which the estimate of θ is unreasonable. This post-processing step is equivalent to establishing a prior on θ in the framework presented in the appendix.

the probability surface. To show that this is not a problem in practice for the point gesture testing EM, we computed the log probability of a testing sequence for all values of θ. This log probability surface, shown in Figure 5, is unimodal, such that for any reasonable initial value of θ the testing EM will converge on the maximum corresponding to the correct value of θ. The probability surfaces of the other test sequences are similarly unimodal.

The peak of the probability surface shown in Figure 5 appears to be a superposition of a mode onto a larger surrounding mode. An examination of the parses recovered at various values of θ shows that points outside of the central peak correspond to parses that are governed almost completely by the Markov model. Equivalently, all state output probabilities are near zero. At points belonging to the peak, however, the state output probabilities are significant and the parse takes a form similar to that of the training sequences.

Lastly, we demonstrate recognition performance of the pointing parameterized HMM. A one minute sequence was collected that contained a variety of movements including six points distributed throughout. To simultaneously detect the gesture and recover θ, we used a 30 sample (one sec) window on the sequence. Figure 7 shows the log probability as a function of time and the value of θ recovered for a number of recovered pointing gestures. All of the pointing gestures were recovered.

5 Non-linear dependencies

When the parameter of interest is a measure of Euclidean distance or coordinates in Euclidean space, the linear model of section 3.1 is appropriate. For situations in which the feature vector is not linear in θ at each state of the HMM, there are at least three courses of action: (1) find an analytical function $f(\theta)$ for which the dependence is linear, and use the new parameters $f(\theta)$ in the place of θ, (2) find some intermediate parameterization that is linear in the feature space and then use some other technique to map to the final parameterization, and (3) use a more general modeling technique, such as neural networks or radial basis function networks.

The first option, for example, would be suited to a model of the pointing motion in which the pointing target is not confined to a limited area in front of the user. In such a case, the target is better represented in spherical coordinates. This is easily done analytically.

The second option involves finding an intermediate parameterization that is linear in the feature space. For example, a musical conductor might convey a dynamic by sweeping out a distance with his or her arm. It may be adequate to model the motion using a parameterized HMM with the distance as the parameter, and then external to the HMM capture some nonlinearity in the mapping from this distance to the intended dynamic by a simple learned function on θ. This technique requires a fine knowledge of how the actual physical movement conveys the quantity of interest.

The last option, employing more general modeling techniques, is naturally suited to situations in which the parameterization is nonlinear and no analytical form of the parameterization is known. For example, when the parameterization is derived from a user's subjective rating it may be difficult or impossible to represent the feature's dependence on θ analytically, especially without some insight as to how the user subjectively rates the motion.

With a more complex model of the dependence on θ (e.g., a neural network), it may not be possible to solve for θ analytically to obtain an update rule for the training or testing EM algorithms. In such a case we may perform gradient descent to maximize Q in the maximization step of the EM algorithm (which would then be called a "generalized expectation-maximization" (GEM) algorithm). In [21] we use neural networks and GEM algorithms to model the subjective quality of a motion.

6 Conclusion

A new method for the representation and recognition of parameterized gesture is presented. The basic idea is to parameterize the underlying output probabilities of the states of an HMM. Because the parameterization is explicit and analytic (here we use a linear relation) the dependence on the parameter θ can be learned within the standard EM formulation.

The method is interesting from two perspectives. First, as a gesture or activity recognition technique it is immediately applicable to scenarios where inputs to be recognized vary smoothly with some meaningful parameter(s). One possible application is advanced human-computer interfaces where the quantity indicating gestures need to be identified and the quantities measured. Also, the technique may be applied to the task of gait recognition, where one would like to ignore the "intensity" of the walk.

Second, the parameterized technique presented is domain independent and is applicable to any sequence parsing problem where some context or style ([**?**]) spans an entire sequence.

As mentioned in the the discussion of non-linear models, it is possible to use non-analytic mappings between θ and the parameters of the output distribution. As long as the mapping is smooth one should be able to do a gradient-based generalized EM maximization step. Investigating this approach is one of our current efforts.

References

[1] A. F. Bobick A. D. Wilson and J. Cassell. Temporal classification of natural gesture and application to video coding. *Proc. Comp. Vis. and Pattern Rec.*, 1997. to appear.

[2] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proceedings of 13th ICPR*, Vienna, Austria, August 1996. IEEE Computer Society Press.

[3] C. M. Bishop. *Neural networks for pattern recognition*. Clarendon Press, Oxford, 1995.

[4] R.A. Brooks and T.O. Binford. Interpretive vision. In *AAAI-80*, pages 21–24, 1980.

[5] M. Svensen C. M. Bishop and C. K. I. Williams. EM optimization of latent-variable density models. In M. C. Moser D. S. Touretzky and M. E. Hasselmo, editors, *Advances in neural information processing systems 8*, pages 402–408. MIT Press, 1996.

[6] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *Proc. Int. Conf. Comp. Vis.*, 1995.

[7] T. Darrell and A. Pentland. Active gesture recognition using partially observable markov decision processes. In *ICPR96*, 1996.

[8] T.J. Darrell and A.P. Pentland. Space-time gestures. *Proc. Comp. Vis. and Pattern Rec.*, pages 335–340, 1993.

[9] Trevor Darrell, Pattie Maes, Bruce Blumberg, and Alex Pentland. A novel environment for situated vision and behavior. In *Proc. of CVPR–94 Workshop for Visual Behaviors*, pages 68–72, Seattle, Washington, June 1994.

[10] W.E.L. Grimson, T. Lozano-Perez, and D.P. Huttenlocher. *Object Recognition by Computer: The Role of Geometric Constraints*. 1990.

[11] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[12] R.E. Kahn and M.J. Swain. Understanding people pointing: The perseus system. In *Proc. IEEE Int'l. Symp. on Comp. Vis.*, pages 569–574, Coral Gables, Florida, November 1995.

[13] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. J. of Comp. Vis.*, 14:5–24, 1995.

[14] S. M. Omohundro. Family discovery. In M. C. Moser D. S. Touretzky and M. E. Hasselmo, editors, *Advances in neural information processing systems 8*, pages 402–408. MIT Press, 1996.

[15] H. Poizner, E. S. Klima, U. Bellugi, and R. B. Livingston. Motion analysis of grammatical processes in a visual-gestural language. In *ACM SIGGRAPH/SIGART Interdisciplinary Workshop, Motion: Representation and Perception*, pages 148–171, Toronto, April 1983.

[16] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[17] J. Schlenzig, E. Hunter, and R. Jain. Vision based hand gesture interpretation using recursive estimation. In *Proc. of the Twenty-Eighth Asilomar Conf. on Signals, Systems and Comp.*, October 1994.

[18] T. E. Starner and A. Pentland. Visual recognition of American Sign Language using hidden markov models. In *Proc. of the Intl. Workshop on Automatic Face- and Gesture-Recognition*, Zurich, 1995.

[19] J. Tennenbaum and W. Freeman. Separating style from content. In *Advances in neural information processing systems 9*, 1997.

[20] A. D. Wilson and A. F. Bobick. Learning visual behavior for gesture analysis. In *Proc. IEEE Int'l. Symp. on Comp. Vis.*, Coral Gables, Florida, November 1995.

[21] A. D. Wilson and A. F. Bobick. (tbd). 1997. (to appear).

A Appendix: Expectation-maximization algorithm for hidden Markov models

In this section we show the derivation of equation 3. We begin by explaining the expectation-maximization (EM) algorithm [3].

EM algorithms are appropriate when there is reason to believe that in addition to the observable data there are unobservable (hidden) data, such that if the hidden data were known, the task of fitting the model would be easier. In the case of HMMs the observable data is the observation sequence $\mathbf{x}_t$, and the hidden data is the state q_t at each time step t. In what follows we denote the entire observation sequence as $\mathbf{x}$ and the entire state sequence as $\mathbf{q}$.

EM algorithms are iterative: the value of the hidden data is computed given the value of some parameters to a model of the hidden and observable data (the "expectation" step), then given this guess at the hidden data, an updated value of the parameters is computed ("maximization"). These two steps are alternated until the change in the overall probability of the observed and hidden data is small (or, equivalently, the change in the parameters is small).

Particular EM algorithms are derived by considering the auxiliary function $Q(\phi' \mid \phi)$, where ϕ denotes the current value of the parameters of the model, and ϕ' denotes the updated value of the parameters. We would like to estimate the values of ϕ'. Q is the expected value of the log probability of the observable and hidden data together given the observables and ϕ:

$$Q(\phi' \mid \phi) = \mathrm{E}_{\mathbf{q}}\left[\log P(\mathbf{x}, \mathbf{q}, \phi') \mid \mathbf{x}, \phi\right] \tag{12}$$

$$= \sum_{\mathbf{q}} P(\mathbf{q} \mid \mathbf{x}, \phi) \log P(\mathbf{x}, \mathbf{q}, \phi') \tag{13}$$

This is the "expectation step". The proof of the convergence of the EM algorithm shows that if during each EM iteration ϕ' is chosen to increase the value of Q (i.e. $Q(\phi' \mid \phi) - Q(\phi \mid \phi) > 0$), then the likelihood of the observed data $P(\mathbf{x} \mid \phi)$ increases as well. The proof holds under fairly weak assumptions on the form of the distributions involved. Choosing ϕ' to increase Q is called the "maximization" step.

Note that if the prior $P(\phi)$ is unknown then we replace $P(\mathbf{x}, \mathbf{q}, \phi') with P(\mathbf{x}, \mathbf{q} \mid \phi')$. In particular, the usual HMM formulation neglects priors on $P(\phi)$. In the work presented in this paper, however, the prior on θ may be estimated from the training set, and furthermore may improve recognition rates, as shown in the results presented in Figure 7.

The parameters ϕ of an HMM include the transition probabilities a_{ij} and the parameters of the output probability distribution associated with each state:

$$Q(\phi' \mid \phi) = \mathrm{E}_{\mathbf{q}} \left[\log \prod_t a_{q_{t-1} q_t} P(\mathbf{x}_t \mid q_t, \phi') \mathbf{x}, \phi \right] \tag{14}$$

The expectation is carried out using the Markov property. $Q(\phi' \mid \phi)$

$$= \mathrm{E}_{\mathbf{q}} \left[\sum_t \log a_{q_{t-1} q_t} + \sum_t \log P(\mathbf{x}_t \mid q_t, \phi') \mid \mathbf{x}, \phi \right] \tag{15}$$

$$= \sum_t \mathrm{E}_{\mathbf{q}} \left[\log a_{q_{t-1} q_t} + \log P(\mathbf{x}_t \mid q_t, \phi') \Big| \mathbf{x}, \phi \right] \tag{16}$$

$$= \sum_{t,j} P(q_t = j \mid \mathbf{x}, \phi) \left[\sum_i P(q_{t-1} = i \mid \mathbf{x}, \phi) \log a_{ij} + \log P(\mathbf{x}_t \mid q_t = j, \phi') \right] \tag{17}$$

In the case of HMM's the "forward/backward" algorithm is an efficient ($O(TN)$, T the length of the sequence, N the number of states) algorithm for computing $P(q_t = j \mid \mathbf{x}, \phi)$.

In the "maximization" step, we compute ϕ' to increase Q. Taking the derivative of equation 17 and writing $P(q_t = j \mid \mathbf{x}, \phi)$ as γ_{tj} we arrive at:

$$\frac{\delta Q}{\delta \phi'} = \sum_t \sum_j \gamma_{tj} \frac{\frac{\delta}{\delta \phi'} P(\mathbf{x}_t \mid q_t = j, \phi')}{P(\mathbf{x}_t \mid q_t = j, \phi')} \tag{18}$$

which we set to zero and solve for ϕ'. In the case of the usual HMM formulation, the familiar Baum-Welch algorithm is obtained.