Decision Theoretic Learning of Human Facial Displays and Gestures

Jesse Hoey and James J. Little
Department of Computer Science
University of British Columbia
Vancouver, BC, CANADA

March 11, 2004

Abstract

We present a vision-based, adaptive, decision-theoretic model of human facial displays and gestures in interaction. Changes in the human face occur due to many factors, including communication, emotion, speech, and physiology. Most systems for facial expression analysis attempt to *recognize* one or more of these factors, resulting in a machine whose inputs are video sequences or static images, and whose outputs are, for example, basic emotion categories. Our approach is fundamentally different. We make no prior commitment to some particular recognition task. Instead, we consider that the *meaning* of a facial display for an observer is contained in its relationship to actions and outcomes. Agents must distinguish facial displays according to their *affordances*, or how they help an agent to maximize utility. To this end, our system learns relationships between the movements of a person's face, the context in which they are acting, and a utility function. The model is a partially observable Markov decision process, or POMDP. The video observations are integrated into the POMDP using a dynamic Bayesian network, which creates spatial and temoral abstractions amenable to decision making at the high level. The parameters of the model are learned from training data using an *a-posteriori* constrained optimization technique based on the expectation-maximization algorithm. One of the most significant advantages of this type of learning is that it does not require labeled data from expert knowledge about which behaviors are significant in a particular interaction. Rather, the learning process *discovers* clusters of facial motions and their relationship to the context automatically. As such, it can be applied to any situation in which non-verbal gestures are purposefully used in a task. We present an experimental paradigm in which we record two humans playing a collaborative game, or a single human playing against an automated agent, and learn the human behaviors. We use the resulting model to predict human actions. We show results on three simple games.

1 Introduction

This paper describes a model of human gestures and facial expressions that unifies computer vision, uncertain reasoning, and decision theory. The motivation is that computational agents will need capabilities for learning, recognising and using the extensive panoply of human non-verbal communication skills. The verb *use* is important: non-verbal signals are *useful*. For example, they may predict the future actions of the signaler, or they may request actions from the perceiver. The perceiver of a non-verbal signal must not only *recognise* the signal, but must *understand* what it is useful for. The signal's usefulness will be defined by its relationship to the joint space of both signaler and receiver, their joint actions, their possible futures together, and the individual ways they assign value to these futures. The model we present in this paper aims to unify the computer vision aspects of automatically perceiving human non-verbal behaviors with the decision-theoretic aspects of putting those perceptions to use. The ability to explicitly reason about uncertainty plays an important role in this unification. If an agent is to take decisions based upon noisy visual data, then the agent must explicitly model its own uncertainty about its perceptions. Bayesian networks are the ideal tool for modeling uncertainty in video measurements and in decision theoretic models, providing a theoretical basis for these models to co-exist.

We claim that it is important not to separate computer vision from decision theory when modeling human behavior. It is not sufficient to build computer vision sensors that deliver information to decision-theoretic reasoning modules, which then decide upon actions. Decision making is difficult, and is best dealt with at a high level of abstraction. Humans seem capable of acting in the world based upon abstract representations of their sensory inputs. Therefore, an efficient perceptual system (which is presumably what humans have evolved to have) will be able to adapt its low-level representation system to only pick up those parts of the signal that are sufficient for building the high-level abstractions. It is important for low-level vision components to not only send, but also receive information from high-level decision-making components. We believe that a unified model of the two aspects is the most effective method for approaching this problem.

The model presented in this paper is a partially observable Markov decision process, or POMDP, with observations over the space of video sequences of human facial displays and gestures. The POMDP model integrates the recognition of the non-verbal signals with their interpretation and use in a utility-maximization framework. The model can be acquired from data, and can be used for decision making based, in part, on the non-verbal behavior of a human through observation. However, optimal decision making in the face of large, continuous, observation spaces is still an open problem, which this paper does not attempt to solve. Instead, we apply a simple approximate solution technique to compute policies of action based on non-verbal displays. The approximate policies work fairly well in the examples we present, but would be insufficient in more complex situations. Finding better approximate (or even optimal?) solutions is part of our current research.

Our work is distinguished from other work on recognising human non-verbal behavior primarily because it does not require labeling of training data for particular facial displays or gestures. We do not train classifiers for different behaviors and then base decisions upon the classifier outputs. Instead, the training process *discovers* categories of behaviors in the data. The advantage of this approach is threefold. First, we do not need expert knowledge about which displays are important, nor extensive human labeling of data. Second, since the system *learns* categories of motions, it will adapt to novel displays without modification, and without *a-priori* specification of said displays. Third, resources can be focused on tasks which will be useful for the agent. It is wasteful to train complex classifiers for the recognition of fine motion if only simple displays are being used in the agent's context. In fact, it is believed that study of this type of active, value-directed learning will enable POMDP solution techniques to scale.

In constrast, psychologists have advocated the use of coding systems for the description of facial action. In particular, the Facial Action Coding System (FACS) [EW78] has become the standard for psychological inquiries into facial expression. Computer vision researchers have also adopted it as the standard to strive towards [EP97, TKC01, BLB+03]. However, although the the recognition of facial action units may give the ability to discriminate between very subtle differences in facial motion, or "microactions", it requires extensive training and domain specific knowledge. For example, Bartlett *et al.* write that it would require approximately 250 minutes, or nearly half a million hand-coded frames of video to train their system for most facial action units [BLB+03], while their current data set contains only 17 minutes of coded video frames. As Pentland has pointed out, the importance of such a fine level of representation is not clear for computer vision systems that intend to take actions based on observations of the human, and the action unit analysis may only be useful for the behavioral sciences [Pen00]. Finally, the same type of analysis is not applicable to gestures or body motion, as these do not have well defined standards of form [McN92]: it will be difficult to come up with a small set of "basic" action units which span the space of all possible gestures.

There will be little discussion of speech recognition and natural language understanding in this paper. Our work focuses solely on recognizing and using non-verbal communicative acts. However, it is well known that gesture and facial expression are intimately tied to speech [McN92,CBC+00], and one might object to our omission. Nevertheless, research has shown that gestures take place *globally* and *synthetically*, as opposed to language, which is linear-segmented (can be broken down in to individual units) [McN92]. Further, semantic gestures do not combine to form larger gestures, but remain one to a clause. Finally, gestures and facial expressions vary from person to person [McN92,Cho91]. These findings give us good reason to keep speech and gesture recognition at a distance, unifying them at some higher level of temporal abstraction.

1.1 Model Design

There has been a growing body of work in the past decade on the communicative function of the face [Fri94,RFD97] and of the hands [McN92]. This psychological research has drawn three major conclusions. First, non-verbal gestures are often purposeful communicative signals [Fri94]. Second, the purpose is not defined by the gesture alone, but is dependent on both the gesture and the context in which the gesture was emitted [RFD97, Cho91]. Third, the signals are not universal, but vary between individuals in their physical appearance, their contextual relationships, and their purpose [RFD97]. We believe that these three considerations should be used as critical constraints in the design of computational communicative agents able to learn, recognise, and use human behavior. That is,

1. Context dependence implies that the agent must model the relationships between the displays and the context in which they are shown.

2. If the agent is to act rationally, then it must be able to compute the utility of taking actions in situations involving purposeful non-verbal displays. It must understand the relationships between the displays, the context, and its own utility function.

3. The signals are individual and context dependent, and so the agent needs to adapt to new interactants and new situations.

These constraints can be integrated in a decision-theoretic, vision-based model of human non-verbal signals. The model can be used to predict human behavior, or to choose actions which maximize expected utility. The basis for this model is a partially observable Markov decision process, or POMDP. A POMDP describes the effects of an agent's actions upon its environment, the utility of states in the environment, and the relationship between the observations, the actions and the states. A POMDP model allows an agent to predict the long term effects of its actions upon his environment, and to choose valuable actions based on these predictions.

1.2 Previous Work

Analysis of human motion involves three major stages. First, the acquisition or segmentation of the part(s) of video to be analysed. For example, detecting and tracking of faces, hands, or entire human bodies is the usual first step towards any facial expression, gesture or human motion recognition system. We use a simple optical flow based tracker, which is not a contribution of this paper. Details can be found in [Hoe04]. The second stage is the extraction of features from the parts acquired in the first stage. The third stage is to classify the extracted features into a discrete set of non-verbal behaviors, such as facial expressions or gestural primitives.

Most of the methods we review will be strictly *supervised*, in that models are trained for one of several pre-defined categories of human behaviors, and are tested for recognition of the same behaviors. In the following, we first cover feature extraction and spatial abstractiton methods, followed by an overview of research on the automatic discovery of human motion patterns from video, and on the purposeful recognition and use of human behavior.

Representation of human behavior in video is usually done by first estimating some quantity of interest at the pixel level, and then spatially abstracting this to a low dimensional feature vector. Optical flow [MP91, BY97, EP97, CT98, LKCL98, DBH+99], color blobs [SP95, Bre97], deformable models [LTC97], motion energy [BD01], and filtered images [BLB+03], are the more well used pixel-level features. This work uses optical flow fields and raw grayscale images, as described in Section 2.1.2 and 2.1.4.

Spatial abstraction is usually approached using either a model-based, or view-based representation of a body part. Model-based approaches are often three dimensional wire-frame models [TK93, BLB+03], sometimes including musculature [EP97]. While model-based approaches can avoid the problems with view-based approaches have with different views of the face or body, these advantages usually come at the cost of computational requirements. View-based approaches spatially abstract video frames by projecting them to a low dimensional subspace [TP91, BY97, BHK97], one of the most well-used begin the principal subspace of variation [Koh89]. Computing this subspace is known as principal components analysis, which has been used to descibe facial structure [TP91], motion in the face [FBYJ00], and spatio-temporal variation in body motion [LF98]. Fisher linear discriminants [BHK97], and independent component analysis [Bar01], are other methods for constructing low dimensional subspaces. Other representations of faces and bodies use templates [BD01, ABMM03], feature points [LTC97, GJH99], or "blobs" [SP95, Bre97].

Our work uses Zernike basis functions [vZ34] for holistic representation of the face and facial motion. The Zernike polynomial basis provides a rich and data independent description of optical flow fields and grayscale images When applied to optical flow, the Zernike basis can be seen as an extension of the standard affine basis [HL00]. The Zernike representation differs from approaches such as Eigen-analysis [TP91], or facial action unit recognition [TKC01] in that it makes no commitment to a particular type of motion, leading to a transportable classification system (e.g., usable for gesture clustering). The Zernike basis has also been used extensively for shape descriptions [Tea80, TC88]. Teh & Chin show that Zernike and pseudo-Zernike moments perform best in terms of noise insensitivity and image reconstruction [TC88]. Zernike polynomials have been used in the vision community for recognising hand poses [HSJ95], handwriting and silhouettes [BSA91], shape-based image retrieval [ZL02], and optical flow fields [HL00].

Once features are computed for each frame, their temporal progression must be modeled. Spatio-temporal templates [EP97], Dynamic time warping [DEP96], and hidden Markov models [SP95] are all popular approaches. Hidden Markov models are a particular case of the more general dynamic Bayesian networks [Pea88, Mur02]. HMMs have been recently been applied to many recognition problems in computer vision, such as hand gestures [SHJ94], and American Sign Language (ASL) [SP95, VM98]. Morimoto, Yacoob and Davis recognised head gestures using HMMs in a view-based approach [MYD96]. Lien *et al.* and Bartlett *et al.* use HMMs to distinguish FACS facial action units [LKCL98, BLB+03].

HMMs are really only the beginning of the story on statistical temporal models, however. They are, in fact, a special case of the more general dynamic Bayesian networks (DBNs), which are simply Bayesian networks in which a discrete time index is explicitly represented. Inference and learning in DBNs is simply an application of network propagation in Bayesian networks [Pea88]. A comprehensive review of DBNs is given by Murphy's thesis [Mur02]. There are many DBN extension of HMMS, including the coupled hidden Markov model [BOP97]. Hierarchical models [FST98, MP01] are particularly interesting, as they incorporate temporal abstractions, and have been used for modeling full body motions [Bre97, PFH99, Bra99, OHG02], and facial expressions [CSG+03]. Our work uses a DBN known as the abstract hidden Markov model, to describe facial expressions [Hoe01].

Most of the methods we have been describing use training processes with labeled data, which requires extensive human intervention, and makes adaptivity more difficult. The alternative is to develop systems that can *discover* categories of motions in training data. In particular, clustering sequences of data using mixtures of hidden Markov models was proposed by Smyth [Smy97], which have been used in computer vision for unsupervised clustering of data [CP99, ASKP03]. Darrell, Essa and Pentland [DEP96] examine the same kind of models, but use dynamic time warping (DTW) instead of hidden Markov models as mixture components. They also use these models for generation of facial expressions. These works do not explicitly model actions and utilities.

Jebara and Pentland [JP98] presented *action-reaction learning*, in which a dynamic model was learned from observing video of two persons interacting. The model was then used in a reactive way to simulate interactions for a single user. The features are color blobs of head and hands, and the joint likelihood of the each person's features is accomplished using a variant of the expectation-maximization algorithm. Our work bears a resemblance to action-reaction learning, but generalizes it by adding high-level context states, actions and utilities. Action-reaction learning is designed strictly for *imitation*-type tasks, while our model is applicable to interactions in more general contexts, in which plans need to be developed autonomously.

However, most of this work takes the slant that the recognition itself is the goal. Clearly, it is what to *do* with the recognised states which is of most interest. Most work in this direction lies between computer vision and human-computer interaction (HCI), in which computer vision systems gather information about the state or actions of a human user of a computer application. This information is used by the application to tailor its interface. Currently, most HCI systems only make use of human interface actions, such as mouse or keyboard actions, some have begun to integrate visual and auditory information [Kje01, LSR+00, Pen00]. In particular, Cassell has stressed the importance of recognising and generating both verbal and non-verbal signals for *embodied conversational agents* (ECAs) [CSPC00]. ECAs are built upon a conversational architecture: they are designed based upon the psychology of human conversational behaviors. Robotic systems are also beginning to make use of computer vision for interactions with humans [FND03, BS99, MPR+02, EHL+02].

The model we focus on in this thesis is the partially observable Markov decision process, or POMDP. POMDPs were applied to the problem of active gesture recognition in [DP96], in which the goal is to model unobservable, non-foveated regions. POMDPs have also been applied to the dialogue management problem [PH00, RPT00, ZCMG01] for human-computer and human-robot interaction. This work, as Cassell's work on ECAs, models some of the basic mechanics underlying dialogue, such as turn taking, channel control, and signal detection. These agents typically use very few (or none at all) manually specified facial expressions or gestures.

2 Non-Verbal Display Understanding using POMDPs

A POMDP is a probabilistic temporal model of an agent interacting with the environment [KLC98], shown as a Bayesian network in Figure 1(a). A POMDP is similar to a hidden Markov model in

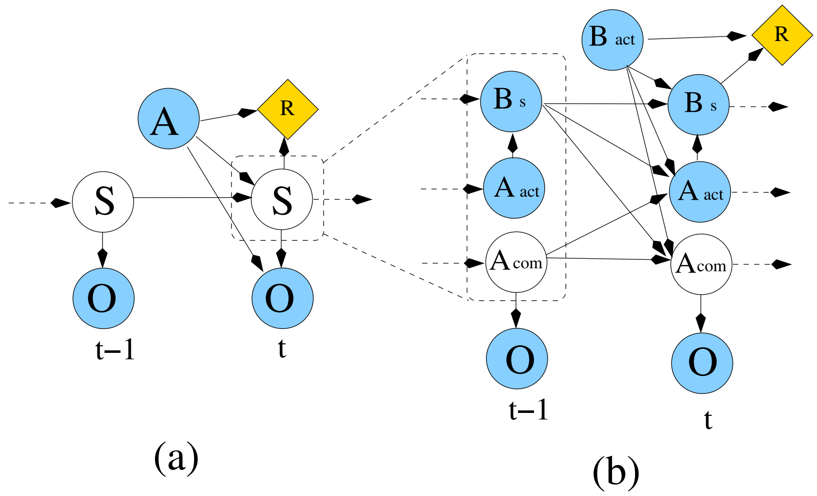


Figure 1: (a) Two time slices of general POMDP. (b) Two time slices of factored POMDP for display understanding. The state, S, has been factored, and conditional independencies have been introduced.

that it describes observations as arising from hidden states, which are linked through a Markovian chain. The Markovian assumption is that the agent's history is contained in its current state. However, the POMDP adds actions and rewards, allowing for decision theoretic planning.

A POMDP is a tuple $\langle S, A, T, R, O, B \rangle$, where S is a finite set of (possible unobservable) states of the environment, A is a finite set of agent actions, $T : S \times A \to S$ is a transition function which describes the effects of agent actions upon the world states, $R : S \times A \to \mathcal{R}$ is a reward function which gives the expected reward for taking action A in state S, O is a set of observations, and $B : S \times A \to O$ is an observation function which gives the probability of observations in each state-action pair. A POMDP model allows an agent to predict the effects of its actions upon his environment, and to choose actions based on its predictions.

To use POMDPs for display understanding, we must admit that the environment may include other intelligent agents, which puts us in the realm of multi-agent games. However, we can take a decision analytic approach to games [Mye91], in which each agent decides upon a strategy based on his subjective assessment of the strategies employed by other players. Essentially, a decision analytic agent includes the strategies and internal states of all other agents as part of his internal state. In general, if the decision analytic agent attributes its partner with decision-making capabilities that are at least as complex as its own, then it must include a complete POMDP as part of its internal state [Gmy02]. We will not examine games with sufficient complexity to warrant this general interactive POMDP.

In the following, we will refer to the two agents we are modeling as "Bob" and "Ann", and we will discuss the model from Bob's perspective. Figure 1(b) shows a factored POMDP model for display understanding in simple interactions [1]. The state of Bob's POMDP is factored into Bob's private internal state, Bs, Ann's action, $Aact$, and Ann's display, $Acom$, such that $S_t = \{Bs_t, Aact_t, Acom_t\}$. While Bs and $Aact$ are observable, $Acom$ is not, and must be inferred from video sequence observations, $\mathbf{O}$. In general, both $Aact$ and Bs may also be unobservable. However, we wish to focus on learning models of displays, $Acom$, and so we will use games in which $Aact$ and Bs are fully observable.

The transition function is factored into four terms. The first involves only fully observable variables, and is the conditional probability of the state at time t under the effect of both player's actions: $\Theta_S = P(Bs_t|Aact_t, Bact, Bs_{t-1})$. The second is over Ann's actions given Bob's action, the previous state, and her previous display: $\Theta_A = P(Aact_t|Bact, Acom_{t-1}, Bs_{t-1})$. The third describes Bob's expectation about Ann's displays given his action, the previous state and her previous display: $\Theta_D = P(Acom_t|Bact, Bs_{t-1}, Acom_{t-1})$. The fourth describes what Bob expects to see in the video of Ann's face, $\mathbf{O}$, given his high-level descriptor, $Acom$: $\Theta_O = P(\mathbf{O}_t|Acom_t)$. For example, for some state of $Acom$, this function may assign high likelihood to sequences in which Ann smiles. This value of $Acom$ is only assigned meaning through its relationship with the context and Bob's action and utility function. We can, however, look at this observation function, and interpret it as an $Acom$ = 'smile' state. For clarity in the following, we rename the variables as $C_t = \{Bact_t, Bs_{t-1}\}$, $A_t = Aact_t$, and $D_t = Acom_t$, the likelihood of a sequence of data, $\{\mathbf{OCA}\}_{1,T} = \{O_1 \ldots O_T, C_1 \ldots C_T, A_1 \ldots A_T\}$, is

$$P(\{\mathbf{OCA}\}_{1,T}|\Theta) = \sum_k \Theta_{O,k} \sum_l \Theta_A \Theta_D P(D_{T-1,l}, \{\mathbf{OCA}\}_{1,T-1}|\Theta)$$

where $\Theta_{O,k}$ is the observation probability given $D_{T,k}$, the k^{th} value of the mixture state, D, at time T. The observations, $\mathbf{O}$, are temporal sequences of finite extent. We assume that the boundaries

[1] Factored representations write the state space implicitly as the cross product of a set of multinomial, discrete variables, and allow conditional independencies in the transition function, T, to be exploited by solution techniques.

of these temporal sequences will be given by the changes in the fully observable context state, C and A. There are many approaches to this problem, ranging from the complete Bayesian solution in which the temporal segmentation is parametrised and integrated out, to specification of a fixed segmentation time [OHG02].

Three parameters in the POMDP model are simple transition matrices: Θ_A, Θ_D and Θ_S. The last parameter, $\Theta_O = P(\mathbf{O}|D)$, is more complex, as it relates spatio-temporally extended observations, $\mathbf{O}$, to high-level behavior descriptors, A. The next section describes this function.

2.1 Observation Function

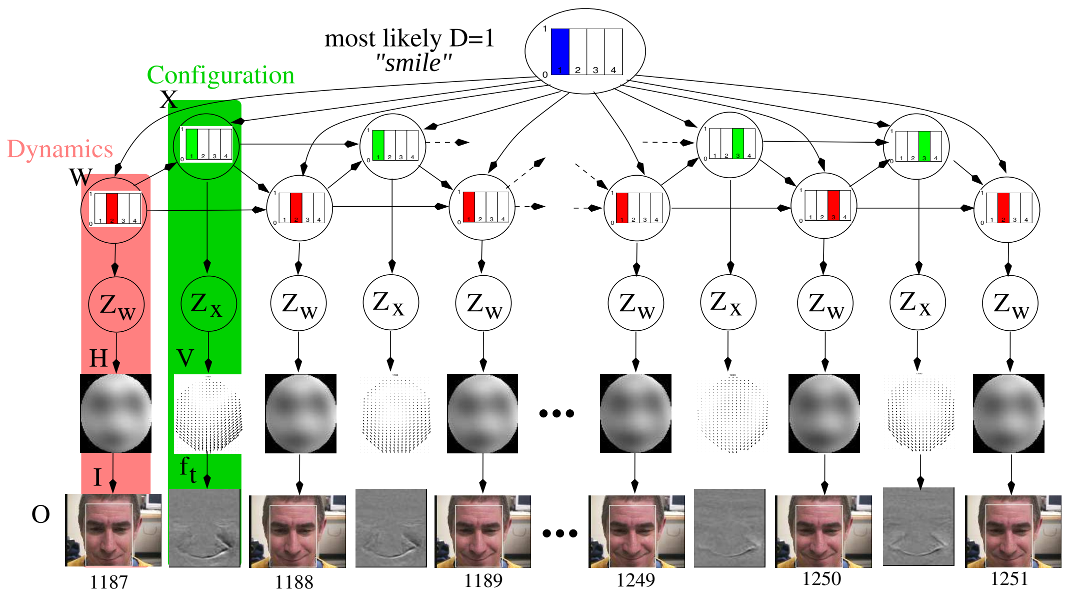


Figure 2: A person smiling is analysed by the mixture of CHMMs. Observations, O, are sequences of images, I, and image temporal derivatives, f_t, both of which are projected over the facial region to a set of basis functions, yielding feature vectors, Z_x and Z_w. The image regions, H, are projected directly, while it is actually the optical flow fields, V, related to the image derivatives which are projected to the basis functions [HL03]. Z_x and Z_w are both modeled using mixtures of Gaussians, X and W, respectively. The class distributions, X and W, are temporally modeled as mixture, S, of coupled Markov chains. The probability distribution over S is at the top. The most likely state, $S = 1$, can be associated with the concept "smile". Probability distributions over X and W are shown for each time step. All other nodes in the network show their expected value given all evidence. Thus, the flow field, v, is actually $\langle v \rangle = \int_v vP(v|O)$.

Figure 2 shows the model as a Bayesian network being used to assess a sequence of a person's hand performing a "stop" gesture. This model is a mixture of coupled hidden Markov models. We consider that spatially abstracting a video frame during a human non-verbal display involves modeling both the current configuration and dynamics of the face. Dynamics and configuration complement one another, and are akin to momentum and position in classical dynamics. They can be both useful in describing the way a human face moves. In particular, modeling the configuration of the face disambiguates temporal sequences of dynamics, while the dynamics of the face can predict future configurations. Configuration and dynamics are both useful for tracking.

Our observations consist of the video image regions, I, and the temporal derivatives, f_t, between pairs of images over these regions. We assume here that the image regions are given at each frame. The temporal derivatives (along with spatial derivatives) induce a dense optical flow field, by assuming that the image intensity structure is locally constant across short periods of time (the *brightness constancy assumption*). The optical flow field is a projection of the 3D scene velocity to the image plane, and gives the motion in the image at each pixel. Thus, the measurements we start from contain simultaneous descriptions of the instantaneous configuration and dynamics of the body. The task is first to spatially summarise both of these quantities, then to temporally compress the entire sequence to a distribution over high level descriptors, D.

The spatial abstraction of images and temporal derivatives occurs in the two vertical chains as shown in Figure 2, culminating in distributions over the multivariate random variables, W and X, for images and temporal derivatives, respectively. W and X correspond to classes of instantaneous configuration and dynamics of the region of interest in the training data. For example, the configuration classes may correspond to characteristic facial poses, such as the apex of a smile. The dynamics classes are motion classes, and may correspond to, for example, motion during expansion of the face to a smile.

The same method is used for spatial abstraction of both the configuration and dynamics of the face. Image regions and optical flow fields are each projected to a pre-determined set of basis functions, yielding finite dimensional feature vectors, Z_w and Z_x, respectively. We use the basis of Zernike polynomials, which have useful properties for modeling flow fields [HL00] and images [TC88]. The distributions of each of the feature vectors (for configuration, Z_w, and dynamics, Z_x) are modeled by a mixture of Gaussians distribution, where the mixture components are labeled as states of W and X. The mixture models at this stage also include feature weights as priors on the cluster means [HL03]. These feature weights obviate the need to choose which basis functions are useful for classification. They are discussed further in Section 2.1.3. In the following, we review Zernike polynomials in Section 2.1.1, then describe the spatial abstraction models (vertical chains in Figure 2), for flow fields and for images in Sections 2.1.2 and 2.1.4, respectively. Finally, we show how the coupled hidden Markov model can temporally abstract a video sequence in Section refsec:chmm.

2.1.1 Zernike Polynomial Basis Functions

Spatial abstraction of flow fields and images involves finding appropriate subspaces in which the motions and poses we are trying to categorize are sufficiently well separated. A standard approach to this problem is to compute a data-dependent subspace using principal components analysis, or PCA [TP91]. PCA methods, however, do not necessarily find the most *useful* subspace, only the one that accounts for the most variability in the data. Further, they require a separate set of basis functions for *every* type of motion we wish to recognize.

We believe that a data *independent* subspace surmounts the two aforementioned problems. We choose a complete and orthogonal set of basis functions *a-priori*, and use them for all our modeling methods. The advantage of data independence is that the basis can equally well be used for representing any motion, without re-computation of a set of basis functions. The usual objection to this type of modeling is that we do not know which set of basis functions are best for a particular modeling task. However, our method includes a feature weighting technique which learns the subset of basis functions which are most useful for the classification task.

Zernike polynomials are an orthogonal set of complex polynomials defined on the unit disk [PR89]. The lowest two orders of Zernike polynomials correspond to the standard affine basis. Higher orders represent higher spatial frequencies. The basis is orthogonal over the unit disk, such that each order can be used as an independent characterization of a 2D function, and each such function has a unique decomposition in the basis. Zernike polynomials are expressed in polar coordinates as a radial function, $R_n^m(\rho)$, modulated by a complex exponential in the angle, ϕ, as follows:

$$U_n^m(\rho,\phi) = R_n^m(\rho)e^{im\phi} \tag{1}$$

with radial function,$R_n^m(\rho)$, given by

$$R_n^m(\rho) = \sum_{l=0}^{(n-|m|)/2} \frac{(-1)^l(n-l)!}{l![\frac{1}{2}(n+|m|)-l]![\frac{1}{2}(n-|m|)-l]!}\rho^{n-2l}$$

for n and m integers with $n \geq |m| \geq 0$ and $n-m$ even.

The indices, n and m, are indicators of the spatial frequency of the Zernike basis function. The larger the value of n, the higher the spatial frequency in the radial direction. Similarly, the larger the value of m, the higher the spatial frequency in the angular direction. For each n, polynomials are defined for a selection of values for m in the range $\{0, n\}$.

The orthogonality of the basis allows the decomposition of an arbitrary function on the unit disk, $f(\rho,\phi)$, in terms of a unique combination of odd and even Zernike polynomials. That is, [PR89]

$$f(\rho,\phi) \approx \sum_{m=0}^{M}\sum_{n=m}^{N}\left[A_n^m cos(m\phi) + B_n^m sin(m\phi)\right]R_n^m(\rho), \tag{2}$$

which can be used to approximate a sufficiently smooth function $f(\rho,\phi)$ to any degree of accuracy by making N and M large enough.

Zernike polynomials are defined on a disk, and so an elliptical area must be identified which will be projected onto the Zernike basis. Although we only consider ellipses with axes aligned with the image axes, it would be possible to allow for rotations of the ellipse in future work. Once a scale and centroid have been identified for each flow image, a 2D function (either a flow field or an image region) $f(x,y)$ is projected onto the Zernike basis using

$$\begin{matrix} A_n^m \\ B_n^m \end{matrix} = \frac{\epsilon_m(n+1)}{\pi}\sum_x\sum_y f(x,y)R_n^m(\rho)\begin{matrix} cos(m\phi) \\ sin(m\phi) \end{matrix} \tag{3}$$

where $\phi = \arctan(y'/x')$, $\rho = \sqrt{x'^2+y'^2} \leq 1$, $x' = (x-x_c)/r_x$, $y' = (y-y_c)/r_y$, $\{x_c, y_c\}$ and $\{r_x, r_y\}$ are the centroid and scales of the region of interest, and $\epsilon_m \equiv 1$ if $m = 0$ or 2 otherwise,

We can write the projection equation (3) for a 2D function $f(x,y)$ as a matrix equation if we write f as a $N \times 1$ column vector by reading pixels from f row-wise from top to bottom, where N is the number of pixels in the function:

$$f = Pz \tag{4}$$

The Zernike basis functions are the columns of the $N \times N_z$ matrix P (also arranged row-wise), and the projection coefficients are in the $N_z \times 1$ column vector z, where N_z is the total number of Zernike polynomial basis functions. The columns of P (and the rows of z) correspond to the Zernike polynomials in order of increasing n and m, alternating bewteen A_n^m and B_n^m, such that columns $0, 1, 2, 3 \ldots$ are Zernike polynomials $A_0^0, A_1^1, B_1^1, A_0^2 \ldots$.

2.1.2 Modeling Facial Dynamics

We wish to classify the instantaneous dynamics of the human face into a discrete set of classes, starting from image derivatives, ∇f. However, the image derivatives by themselves are not sufficient to describe image motion, because there is a many-to-one correspondence between derivatives and motion. We can constrain the derivatives using the hypothesis that the intensity structure of the scene is locally stable across short time intervals [HS81]. This allows us to estimate the way things are moving, or the *optical flow*, in the image plane between frames, which is what we want to classify.

If some small part of the world appears at time t at some position in the image, then we assume it will appear the same at $t+1$, albeit in a different position in the image. The difference in position is the (true) optical flow, v, and the relationship is known as the *brightness constancy assumption*.

$$I(x, y, t) \approx I(x + v_x \delta t, y + v_y \delta t, t + \delta t) \tag{5}$$

Simoncelli [SAH91] has described how to write this equation in a Bayesian framework, in which the variability due to error sources is explicitly included in the model. We describe noise as arising from two independent zero-mean Gaussian noise sources, n_1 and n_2, which account for failures of the planarity assumption, and errors in the temporal derivative measurements, respectively. The *brightness constancy* assumption thus becomes [SAH91]

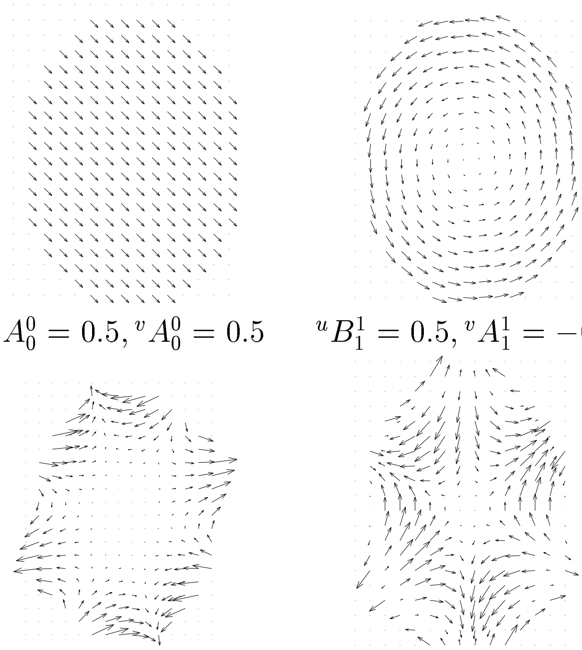
$$f_\tau + f_s \cdot (v - n_1) = n_2, \qquad n_i \sim \mathcal{N}(0, \Lambda_i). \tag{6}$$

We can assume that the errors in the spatial derivatives are minimal compared to those in the temporal derivatives, since the temporal sampling is much coarser than the spatial sampling. Thus, $P(\nabla f|v) = P(f_t|v, f_s)$, and Equation 6 describes the conditional probability

$$P(\nabla f|v) \propto \mathcal{N}(f_\tau; -f_s v, f_s \Lambda_1 f_s' + \Lambda_2), \tag{7}$$

where $\Lambda_1 = \sigma_1 I_N$, $\Lambda_2 = \sigma_2 I_N$ (I_N is $N \times N$ identity). The important thing to notice about this distribution is the dependence of the variance on the spatial derivative, f_s. The magnitude of the spatial derivative, $\|f_s\|^2$, is the image contrast, which plays an important role in determining the distribution of flow fields [SAH91]. Optical flow is difficult to estimate (and so has high variance) in regions of low contrast.

Equation 7 can be used to estimate optical flow directly using a zero-mean prior on the optical flow fields and Bayes' rule [SAH91]. We refer to this optical flow estimation technique as the Simoncelli method in the following. However, we are concerned in this work with *interpreting* the optical flow field as a distribution over a small, temporally and spatially abstract set of discrete states. Classification of optical flow fields directly is difficult due to the high dimensionality of the signal. Instead, we classify optical flow in a subspace of flow fields defined by their projections to the Zernike polyomial basis.



${}^{u}A_0^0 = 0.5, {}^{v}A_0^0 = 0.5$ $\quad$ ${}^{u}B_1^1 = 0.5, {}^{v}A_1^1 = -0.5$

${}^{u}A_3^3 = 0.5, {}^{u}B_3^3 = -1.0, {}^{v}A_3^3 = -0.5$ $\quad$ ${}^{u}B_2^4 = 1.0, {}^{v}A_2^4 = 1.0, {}^{v}B_2^4 = -0.5$

Figure 3: Example flows generated from ZPs corresponding to the indicted subsets of the feature dimensions.

The lowest orders of the Zernike basis, when used to describe optical flow fields, corresponds to the affine basis, which is capable of representing simple planar motions such as translations, rotations, and expansions. The next order polynomials correspond to extensions of the affine basis, roughly *yaw*, *pitch* and *roll*. In particular, Black & Yacoob found these next orders to be particularly useful for modeling motion around the mouth region [BY97].

Equation 3 applied to the horizontal flow field, $v_x(x, y)$, gives the projection coefficients, ${}^{u}A_n^m$ and ${}^{u}B_n^m$:

$$\begin{matrix} {}^{u}A_n^m \\ {}^{u}B_n^m \end{matrix} = \frac{\epsilon_m(n+1)}{\pi} \sum_x \sum_y v_x(x, y) R_n^m(\rho) \begin{matrix} cos(m\phi) \\ sin(m\phi) \end{matrix} \tag{8}$$

A similar set of equations is obtained for the vertical flow estimates, ${}^{v}A_n^m$ and ${}^{v}B_n^m$. Figure 3 shows some example flow fields reconstructed from different orders (values of n and m) of Zernike polynomials. Higher orders of Zernike polynomials result in flow fields with higher spatial frequencies, representing more complex motions. The flows can be reconstructed from the coefficients using Equation 3. As we reconstruct with more coefficients, we are including higher spatial frequencies, leading to a more accurate reconstruction of the original.

As in Equation 4, we can represent the optical flow field projections as $v = Mz$, where

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad M = \begin{bmatrix} P & 0 \\ 0 & P \end{bmatrix} \quad z = \begin{bmatrix} z_x \\ z_y \end{bmatrix}. \tag{9}$$

The columns of P are the N_z basis vectors and z_x, z_y are the Zernike coefficients for horizontal and vertical flow, respectively, $\{{}^uA_n^m, {}^uB_n^m\}$ and $\{{}^vA_n^m, {}^vB_n^m\}$. In practice, M will be some subset of the Zernike basis vectors, the remaining variance in the flow fields being attributed to zero-mean Gaussian noise. Thus, we write $v = Mz + n_p$, where $n_p \propto \mathcal{N}(0, \Lambda_p)$, and so

$$P(v|z) = \mathcal{N}(v; Mz, \Lambda_p) \tag{10}$$

The noise, n_p, is a combination of three noise sources: the reconstruction error (energy in the higher order moments not in M), the geometric error (due to discretization of a circular region), and the numerical error (from discrete integration) [LP98].

We wish to use these flow field projections for classification tasks, in which some flow field, v, is classified as originating from one of a set of causes, X. Figure 4 shows the model represented as a Bayesian network. This is a detailed version of the dynamics vertical chain from Figure 2.

We can express classification of an image motion as the maximization of the probability distribution over the classes, X, given the spatial and temporal derivatives,

$$P(X|\nabla f, \Theta) \propto P(\nabla f|X, \Theta)P(X|\Theta), \tag{11}$$

where Θ are the parameters of the model, and $\nabla f = \{f_x, f_y, f_t\}$. Since we wish to classify optical

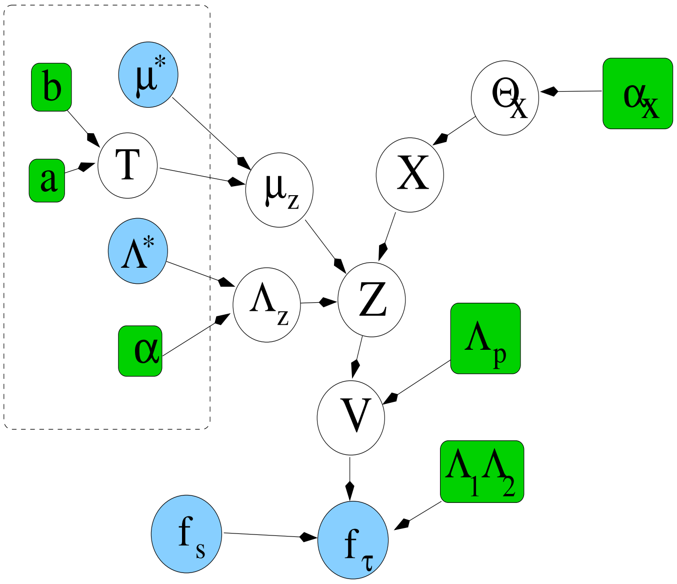


Figure 4: Bayesian network for the mixture of Gaussians over optical flow fields with feature weighting. Shaded nodes are observed or fixed (known), while unshaded nodes are unknown random variables. Boxes are fixed hyper-parameters. The dashed line delineates the priors for feature weighting. $X \in 1 \ldots N_x$ are discrete motion classes, Z is the Zernike feature vector (projection of optical flow field), V is the optical flow field, f_s are the spatial derivatives, and f_t is the temporal derivative. μ_z, Λ_z are the parameters of the mixture of Gaussians over the Z vector space, and T are the feature weights. Θ_X are the class probability parameter (a multinomial), and α_X is the parameter of the (conjugate) Dirichlet prior over Θ_X.

flow fields, we expand the probability distribution over the classes, X, as

$$P(X|\nabla f, \Theta) = \int_v P(\nabla f|v, \Theta)P(v|X, \Theta)P(X|\Theta)$$

where we have assumed the image derivatives to be independent of the high level motion class given the optical flow.

There are three terms in the integration. The prior over classes, $P(X|\Theta)$, is part of our model, parametrized with a multinomial $\Theta_{x,i} = P(X = i)$. The distribution over spatio-temporal derivatives conditioned on the flow, $P(\nabla f|v, \Theta)$, is estimated in a gradient-based formulation using the brightness constancy assumption, and is given by Equation 7.

We do not represent $P(v|X)$ directly in our model, but instead we parametrise this distribution using a probabilistic projection of v to the basis of Zernike polynomials. As shown in Equation 10, this projection can be written as a distribution over v, given the projection coefficients, z, $P(v|z) \propto \mathcal{N}(v; Mz, \Lambda_p)$. We then parametrise the distribution over z given X with a normal $P(z|X) = \mathcal{N}(z; \mu_{z,x}, \Lambda_{z,x})$. We are expecting flow fields to be normally distributed in the space of the basis function projections.

We can now write down the likelihood of the image derivatives given the high-level motion class as

$$P(f_\tau|X, f_s, \Theta) = \int_{v,z} \mathcal{N}(f_\tau; -f_s v, A)\mathcal{N}(v; Mz, \Lambda_p)\mathcal{N}(z; \mu_{z,x}, \Lambda_{z,x}) \tag{12}$$

where $A = f_s \Lambda_1 f_s' + \Lambda_2$. Since all terms in Equation 12 are Gaussian distributions, we can perform the integrations over v and z analytically by successively completing the squares in v and z to obtain

$$P(f_\tau|X f_s) = \frac{\sqrt{|\tilde{\Lambda}_{z,x}|}}{\sqrt{|A||\Lambda_{z,x}|}} e^{\frac{1}{2}(\tilde{\mu}_{z,x}' \tilde{\Lambda}_{z,x}^{-1} \tilde{\mu}_{z,x} - \mu_{z,x}' \Lambda_{z,x}^{-1} \mu_{z,x} - \epsilon)} \tag{13}$$

where

$$\begin{aligned}
\Lambda_w &= (f_s' A^{-1} f_s + \Lambda_p^{-1})^{-1} \\
\tilde{\Lambda}_{z,x} &= (\Lambda_{z,x}^{-1} + M'(\Lambda_p + (f_s' A^{-1} f_s)^{-1})^{-1} M)^{-1} \\
\tilde{\mu}_{z,x} &= \tilde{\Lambda}_{z,x}(\Lambda_{z,x}^{-1} \mu_{z,x} - M' \Lambda_p^{-1} \Lambda_w w) \\
\epsilon &= f_\tau' A^{-1} f_\tau + w' \Lambda_w w \qquad w = f_s' A^{-1} f_\tau
\end{aligned} \tag{14}$$

If we normalize this distribution over X, we can remove all terms which are independent of X, and obtain

$$\frac{P(f_\tau|X f_s)}{\sum_x P(f_\tau|X f_s)} = \frac{\sqrt{|\tilde{\Lambda}_{z,x}|}}{\sqrt{|\Lambda_{z,x}|}} e^{\frac{1}{2}(\tilde{\mu}_{z,x}' \tilde{\Lambda}_{z,x}^{-1} \tilde{\mu}_{z,x} - \mu_{z,x}' \Lambda_{z,x}^{-1} \mu_{z,x})}. \tag{15}$$

The mean, $\tilde{\mu}_{z,x}$, and covariance, $\tilde{\Lambda}_{z,x}$, are the parameters of the distribution of basis vector coefficients, z:

$$P(z|X\nabla f) = 2\pi^{-\frac{N_z}{2}} |\tilde{\Lambda}_{z,x}|^{-\frac{1}{2}} e^{-\frac{1}{2}(z-\tilde{\mu}_{z,x})' \tilde{\Lambda}_{z,x}^{-1} (z-\tilde{\mu}_{z,x})},$$

The expected value of Z given the entire model is

$$\tilde{z} = \int_z z P(z|\nabla f) = \sum_i \tilde{\mu}_{z,i} P(x_i|\nabla f). \tag{16}$$

The distribution over X in this expression is computed as

$$P(x_i|\nabla f) = \frac{P(\nabla f|x_i)\Theta_{x,i}}{P(\nabla f)}$$

where $P(\nabla f|x_i)$ is given by Equation (13), and $P(\nabla f)$ normalizes the distribution. The expected flow field, $\tilde{v}$, for a given state, $\tilde{v}_x$, and for the whole model, $\tilde{v}$, can be computed as

$$\tilde{v}_x = M\tilde{\mu}_{z,x} \qquad \tilde{v} = M\tilde{z} \tag{17}$$

The brightness constancy assumption fails if the velocity v is large enough to produce aliasing. Therefore, a multi-scale pyramid decomposition of the optical flow field must be used. This results in distribution over the flow vectors, $P(v|\nabla f) \sim \mathcal{N}(v; \mu_v, \Lambda_v)$, where $\Lambda_v = (f_s' A^{-1} f_s)^{-1}$ and $\mu_v = -\Lambda_v f_s' A^{-1} f_\tau$ [SAH91]. Using these coarse-to-fine estimates, Equations 14 become

$$\begin{aligned} \tilde{\Lambda}_{z,x} &= (\Lambda_{z,x}^{-1} + M'(\Lambda_p + \Lambda_v)^{-1} M)^{-1} \\ \tilde{\mu}_{z,x} &= \tilde{\Lambda}_{z,x}(\Lambda_{z,x}^{-1}\mu_{z,x} + M'(\Lambda_p + \Lambda_v)^{-1}\mu_v) \end{aligned} \tag{18}$$

The mean of this distribution, $\tilde{\mu}_{z,x}$, is a weighted combination of the mean Zernike projection from the data ($M'\mu_v$), and the model mean, $\mu_{z,x}$.

2.1.3 Feature Weighting

In general, we will not know which basis coefficients are the most useful for our classification task: which basis vectors should be included in M, and which should be left out (as part of n_p). We use the feature weighting techniques of [CdFGT03], which characterize the relevance of basis vectors by examining how the cluster means, $\mu_{z,x}$, are distributed along each basis dimension, $k = 1 \ldots N_z$. Relevant dimensions will have well separated means (large inter-class distance along that dimension), while irrelevant dimensions will have means which are all similar to the mean of the data, μ^*.

To implement these notions, we place a conjugate normal prior on the cluster means, $\mu_{z,x} \sim \mathcal{N}(\mu^*, T)$, where T is diagonal with elements $\tau_1^2 ... \tau_{N_z}^2$, and τ_k^2 is the feature weight for dimension k. The prior biases the model means to be close to the data mean along dimensions with small feature weights (small variance of the means), but allows them to be far from the data mean along dimensions with large feature weights (large variance of the means). Thus, τ_k^2 will be large if k is a dimension relevant to the clustering task, while $\tau_k^2 \to 0$ if the dimension is irrelevant. Feature selection occurs if we allow $\tau_k^2 = 0$ for some k.

Conjugate priors are placed on the feature weights, τ_k^2, and on the model covariances, $\Lambda_{z,x}$. Each feature weight is univariate, and so an inverse gamma distribution is the prior on each τ_k^2:

$$P(\tau_k^2|a, b) \propto (\tau_k^2)^{-a-1} e^{-b/\tau_k^2}. \tag{19}$$

This prior allows some control over the magnitude of the learned feature weights, τ_k^2. The model covariances are multivariate, for which the conjugate prior is an inverse-Wishart prior:

$$P(\Lambda_{z,x}|\alpha, \Lambda^*) \propto |\Lambda_{z,x}|^{-(\alpha+N_z+1)/2} e^{-\frac{1}{2} tr(\alpha \Lambda^* \Lambda_{z,x}^{-1})}, \tag{20}$$

where Λ^* is the covariance of all the data, and α is a parameter which dictates the expected size of the clusters (the intra-class distance). This prior stabilizes the cluster learning.

2.1.4 Modeling Facial Configuration

The classification of image configurations is to assign each of a set of images, $I_1 \dots I_{N_t}$, to one of N_w cluster labels $W_1 \dots W_{N_w}$. We use the same model described in the last section, as shown in Figure 5, which is the same as Figure 4, except the labels have changed. Again, this is a detailed version of the vertical configuration chain in Figure 2. The measurements are now the images, I. The subspace projections over image regions are labeled H. We can express classification of an image as the maximization of the probability distribution over the classes, W, given the image:

$$P(W|I,\Theta) \propto P(I|W,\Theta)P(W|\Theta), \tag{21}$$

where Θ are the parameters of the model. Since we plan to classify image projections, we expand

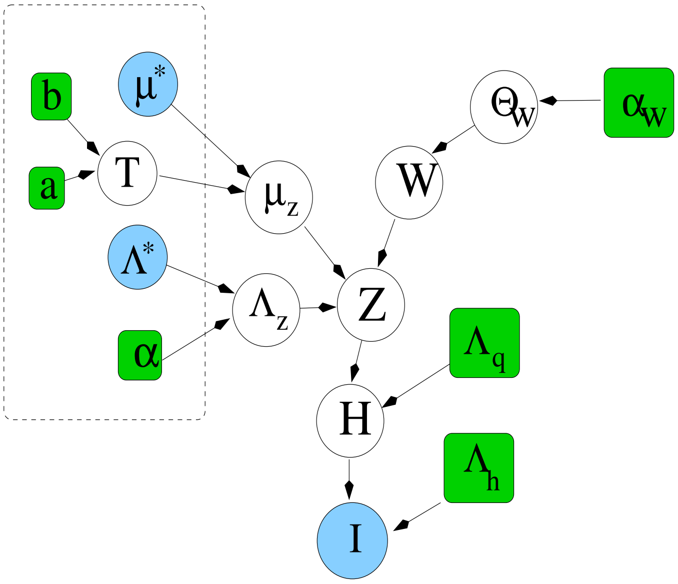


Figure 5: Bayesian network for the mixture of Gaussians over images with feature weighting. Shaded nodes are observed or fixed (known), while unshaded nodes are unknown random variables. Boxes are fixed hyper-parameters. The dashed line delineates the priors for feature weighting. $W \in 1 \dots N_w$ are discrete image classes, Z is the Zernike feature vector (projection of image), H is the projected image region, and I is the full image. μ_z, Λ_z are the parameters of the mixture of Gaussians over the Z vector space, and T are the feature weights. Θ_W are the class probability parameter (a multinomial), and α_W is the parameter of the (conjugate) Dirichlet prior over Θ_W.

this probability distribution as

$$P(W|I,\Theta) = \int_{h,z} P(I|h,\Theta)P(h|z,\Theta)P(z|W\Theta)P(W|\Theta)$$

There are four terms in the integration. The prior over classes, $P(W|\Theta)$, is part of our model, parametrized with a multinomial $\Theta_{w,i} = P(W = i)$. The distribution over z given W is parametetrised with a normal $P(z|W) = \mathcal{N}(z; \mu_{z,w}, \Lambda_{z,w})$. We again use the Zernike polynomial basis to describe image regions, and so, as in the dynamics case, the distribution over the image

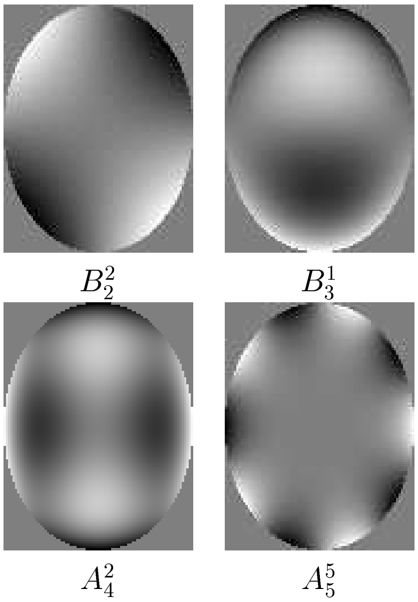


Figure 6: Example images generated from individual ZPs as shown.

regions given the Zernike projection is normal. As with the flow fields, we can write $H = Pz + n_q$, where $n_q \propto \mathcal{N}(0, \Lambda_q)$, the columns of P are the N_z basis vectors and z are the Zernike coefficients, A_n^m and B_n^m. This defines the likelihood $P(H|z) = \mathcal{N}(H; Pz, \Lambda_q)$. The same feature weighting technique is used to estimate the relevance of each dimension of z. We use a small subset (only the first 32 basis polynomials), which gives a very coarse approximation to the brightness structure over the face. Figure 6 shows some examples of Zernike polynomials of different orders (values of n and m) as grayscale images. Higher orders of Zernike polynomials represent more complex brightness patterns.

The distribution over images given the subspace image region, h, $P(I|h, \Theta)$, can be approximated using a normal distribution at each pixel, $P(I|h, \Theta) \sim \mathcal{N}(I; h, \Lambda_h)$. In this case, however, there are no data-dependent variances as in the flow field case. Therefore, the naïve projection is a good approximation to the full integration, and we write

$$P(I|W\Theta) = P(z_W|W\Theta), \tag{22}$$

where $z_W = M'I$ is the projection of the image region to the basis set.

2.1.5 Temporal abstraction using CHMMs

The dynamics and configuration variables, X and W, each form Markovian chains, called the *dynamics* and *configuration* processes, which are coupled, as shown in Figure 2. Temporal abstraction is achieved using a mixture model at the high level, where the mixture components, D, are coupled hidden Markov models. This mixture model can be used to compute the likelihood of a video sequence given the facial display descriptor, $P(\mathbf{O}|D)$:

$$P(\{\mathbf{O}\}_{1,T}|D) = \sum_{ij} \Theta_f \Theta_I \Theta_{Xijk} \sum_{kl} \Theta_{Wjkl} P(X_{T-1,k}, W_{T-1,l} \{\mathbf{O}\}_{1,T-1}|D) \tag{23}$$

where $\Theta_f = P(f_t|X_{T,i})$ and $\Theta_I = P(I_t|W_{T,j})$ are given by Equations 13 and Equation 22, respectdively. The transition functions in the coupled chains are Θ_{Xijk} and Θ_{Wjkl}:

$$\theta_{Wjkl} = P(W_t = j|W_{t-1} = k, X_{t-1} = l) \qquad \Theta_{Xijk} = P(X_t = i|X_{t-1} = jW_t = k).$$

3 Learning POMDPs

The preceding section described a method for representing flow fields and poses over a discrete set of normal distributions. It showed how to combine the modeling of facial dynamics and configurations in a temporal model, and how to build temporal abstractions of the sequences, leading to descriptions of entire sequences of motions and poses over extended periods of time. It also demonstrated high level observable context and actions can be included. In this section, we show how to learn the parameters of these models from data.

This problem can be formulated as a constrained optimization of the likelihood of the observations given the model, over the (constrained) model parameters. We use the well-known *expectation-maximization*, or EM, algorithm to find a locally optimal solution [DNR77]. If we can find a good initialization to the model, then the EM algorithm will optimize this model locally.

3.1 Learning POMDP Parameters

It is important to stress that the learning takes place over the *entire* model simultaneously: both the output distributions, including the mixtures of coupled HMMs, and the high-level POMDP transition functions are all learned from data during the process. The learning classifies the input video sequences into a spatially and temporally abstract finite set, *Acom*, and learns the relationship between these high-level descriptors, the observable context, and the action. Learning the POMDP parameters is to find the set of parameters, $\mathbf{\Theta}^*$, which maximize the posterior density of all observations and the model, $P(\mathbf{OCA\Theta})$, subject to constraints on the parameters. The EM algorithm eases this maximization by writing it as

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} \Big[\sum_{\mathbf{D}} P(\mathbf{D}|\mathbf{OCA\Theta}') \log P(\mathbf{DOCA}|\mathbf{\Theta}) + \log P(\mathbf{\Theta}) \Big]$$

The "E" step of the EM algorithm is to compute the expectation over the hidden state, $P(\mathbf{D}|\mathbf{OCA\Theta}')$, given $\mathbf{\Theta}'$, a current guess of the parameter values. The "M" step is then to perform the maximization which, in this case, can be computed analytically by taking derivatives with respect to each parameter, setting to zero and solving for the parameter. The resulting update equations for the parameters of the POMDP transition functions are the same as for an *input-output* hidden Markov model [BF96].

The update equation for the D transition parameter, $\Theta_{Dijk} = P(D_{t,i}|D_{t-1,j}C_{t,k})$, is then

$$\Theta_{Dijk} = \frac{\alpha_{Dijk} + \sum_{t\in\{1\ldots N_t\}|C_t=k} P(D_{t,i}D_{t-1,j}|\mathbf{O},\mathbf{A},\mathbf{C}\theta')}{\sum_i \left[\alpha_{Dijk} + \sum_{t\in\{1\ldots N_t\}|C_t=k} P(D_{t,i}D_{t-1,j}|\mathbf{O},\mathbf{A},\mathbf{C}\theta')\right]}$$

where the sum over the temporal sequence is only over time steps in which $C_t = k$, and α_{Dijk} is the parameter of the Dirichlet smoothing prior. The summand can be factored as

$$P(D_{t,i}D_{t-1,j}|\mathbf{O}, \mathbf{A}, \mathbf{C}\theta') = \beta_{t,i}\Theta_{A*i*}P(\mathbf{O}_t|D_{t,i})\Theta_{Dijk}\alpha_{t-1,j}$$

where $\alpha_{t,j} = P(D_{t,j}\{\underset{1,t}{\mathbf{OAC}}\})$ and $\beta_{t,i} = P(\{\underset{t+1,T}{\mathbf{OAC}}\}|D_{t,i})$ are the usual forwards and backwards variables, for which we can derive recursive updates

$$\alpha_{t,j} = \sum_k P(\mathbf{O}_t|D_{t,j})\Theta_{A*j*}\Theta_{Djk*}\alpha_{t-1,k} \qquad \beta_{t-1,i} = \sum_k \beta_{t,k}\Theta_{A*k*}P(\mathbf{O}_t|D_{t,k})\Theta_{Dki*}$$

where we write $\Theta_{A*j*} = P(A_t = *|D_{t,j}C_t = *)$ and $P(\mathbf{O}_t|D_{t,i})$ is the likelihood of the data given a state of the mixture of CHMMs (Equation 23). The updates to $\Theta_{Aijk} = P(A_{t,i}|D_{t,j}C_{t,k})$ are $\Theta_{Aijk} = \sum_{t\in\{1...N_t\}|A_t=i\vee C_t=k} \xi_j$, where $\xi_j = P(D_{t,j}|\mathbf{OAC}) = \beta_{t,j}\alpha_{t,j}$. The updates to the j^{th} component of the mixture of CHMMs is weighted by ξ_j, but otherwise is the same as for a normal CHMM [BOP97]. Evidence is propagated backwards and forwards through both X and W chains. Of course, the CHMM can be considered as a simple HMM by considering a flat representation of the factored state, $Y = \{X, Y\}$. The updates to the output distributions in the configuration process, $P(I|W)$, and in the dynamics process, $P(\nabla f|X)$, are as they would be in a mixture model, except that the feature weights bias the updates towards the prior distributions. The update equation for the mean of the i^{th} Gaussian output distribution (a component of $P(z|X)$), $\mu_{z,i}$, is

$$\mu_{z,i} = (\xi_{\cdot,i}\Lambda_{z,i}^{-1} + T^{-1})^{-1}\left[\Lambda_{z,i}^{-1}\left(\sum_{k=1}^{N_t}\tilde{\mu}_{z,x}\xi_{k,i}\right) + T^{-1}\mu^*\right]$$

where $\xi_{k,i} = P(X_{k,i}|\nabla f_k\Theta')$, $\xi_{\cdot,i} = \sum_{k=1}^{N_t}\xi_{k,i}$ and $\tilde{\mu}_{z,x}$ is given by Equation (18). Thus, the most likely mean for each state x is the weighted sum of the most likely values of z as given by Equation (18). Dimensions of the means, $\mu_{z,i}$, with small feature weights, τ_k^2, will be biased toward the data mean, μ^*, in that dimension. This is reasonable, because such dimensions are not relevant for clustering, and so should be the same for any cluster, X.

The updates to the feature weights are

$$\tau_k^2 = \frac{b}{a + N_x/2 + 1} + \frac{1}{2a + N_x + 2}\sum_{i=1}^{N_x}(\mu_{z,i,k} - \mu_k^*)^2$$

and show that those dimensions, k, with $\mu_{z,i,k}$ very different from the data mean, μ_k^*, across all states, will receive large values of τ_k^2, while those with $\mu_{z,i,k} \sim \mu_k^*$ will receive small values of τ_k^2. Intuitively, the dimensions along which the data is well separated (large inter-class distance) will be weighted more. The complete derivation, along with the updates to the output distributions of the CHMMs, including to the feature weights, can be found in [Hoe04].

3.1.1 Initialization

The EM algorithm performs hill climbing on the likelihood surface, converging from its starting point to a local maximum. It is therefore dependent on the initial choice of the parameters. In models with many parameters as we have described, the likelihood surface can contain many local maxima. It is therefore critical to achieve good initialization, introducing as much prior knowledge about the domain before attempting a full maximization of the posterior probability of the model given the data.

Initialization proceeds in a bottom-up fashion in the model. First, the mixture models (vertical chains in Figure 2) are initialized. The dynamics mixture model with N_x classes is initialized from a set of single (independent) spatio-temporal derivative fields, $\nabla \mathbf{f}$, by first computing the expected most likely values of Z for each frame using single a zero-mean model with constant diagonal covariance 0.001, and then fitting a Gaussian mixture to the result of K-means clustering with $K = N_x$. While the K-means algorithm uses the Euclidean distance in the space of Z, the Gaussian fits use the Mahalanobis distance. All the feature weights, τ_k^2, as initialized to 1 and state assignment probability Θ_X is initialized evenly. The mixture model over the configurations is initialized in a similar way, using the projections of image regions to the Zernike basis.

The entire model is then initialized using these estimates of dynamics and configuration mixture models.

1. Classify all the data (including the data that did not pass the thresholding test, above) using the two mixture models. Find the set of X states visited by each sequence.

2. Find the largest N_d sets of sequences whose sets of visited X states match exactly.

3. Find the set of X states visited by all the sequences in each cluster, i. This gives the number of X states for the dynamics chain of model i, N_X^i. Do the same for the W states, giving N_W^i.

4. Initialize a coupled hidden Markov model for each cluster, i, by assigning the output distributions (including feature weights, if applicable) to be those in the simple mixture models which are used by the sequences in the cluster. Initialize the transition and initial state probabilities randomly.

5. Train each coupled hidden Markov model, keeping the output distributions fixed, and initialize the mixture probabilities for the mixture of coupled HMMs evenly for each context state.

Smyth [Smy97] suggests a different initialization method for mixtures of hidden Markov models, which fits a simple HMM to each individual sequence, evaluates the log-likelihood of each sequence given every simple HMM, and then clusters the sequences into K groups using the log-likelihood distance matrix. Simple HMMs are then fit to each of these K clusters, and the results are used to initialize the final mixture model. However, our experiments have shown that the resulting models tend to be significantly “washed out”, and do not find clusters which are well matched with the context states. The reason is that the individual HMMs are not sufficiently well supported by the data, and tend to learn models heavily biased by the prior distributions. We have chosen the method described above to take advantage of the good initializations that can be performed at the lowest level (the Gaussian and multinomial output distributions). This level is very important since it involves many parameters and performs the spatial abstraction step which is crucial to the efficient description of our data.

3.2 Solving POMDPs

If observations are drawn from a finite set, then an optimal policy of action can be computed for a POMDP [KLC98] using dynamic programming over the space of the agent's belief about the state, $b(s)$. However, if the observation space is continuous, as in our case, the problem becomes much more difficult. In fact, there are no known algorithms for computing optimal policies for such problems. Nevertheless, approximation techniques have been developed [BDH99], the simplest of which simply considers the POMDP as a fully observable MDP (the *MDP approximation*): the state, S, is assigned its most likely value in the belief state, $S = \arg\max_s b(s)$. This approximation discards all the probability distributions that the output model has produced. However, less approximate solution techniques ... This approximation will be sufficient for the examples we present. Dynamic programming updates consist of computing value functions, V^n, where $V^n(s)$ gives the expected value of being in state s with a future of n stages to go, assuming the optimal actions are taken at each step. The actions that maximize V^n are the policy with n stages to go. n is also known as the *horizon.* These value functions are computed by setting $V^0 = R$ (the reward function), and then iterating [KLC98]

$$V^{n+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left\{ \sum_{t \in \mathcal{S}} Pr(t|a, s) \cdot V^n(t) \right\} \tag{24}$$

The actions that maximize Equation 24 form the approximately optimal n stage-to-go policy, $\pi^n(s)$. It is also possible to let $n \to \infty$, by also including a discount factor, β. We will only consider finite horizon policies in this paper, however.

3.3 Value directed structure learning

The value function, $V(s)$, gives the expected value for the decision maker in each state. However, there may be parts of the state space which are indistinguishable (or nearly so) with respect to certain characteristics, such as value or optimal action choice. These indistinguishable states can be grouped or merged together to form an *aggregate* or *abstract* state. The set of abstract states *partitions* the state space according to some characteristic. States of the original MDP which are part of the same abstract state are not distinguishable insofar as decisions go. Eliminating the distinctions between them by merging states can lead to efficiency gains without compromising decision quality. An agent needs only distinguish those states which are useful to it for achieving value.

In fact, such state aggregation is a form of structure learning based upon the utility of states. The idea is that a perceptual agent need only make those distinctions which are necessary for predicting future reward. While this idea has been explored in the machine learning litterature [McC93], this paper shows how it can be used in a realistic domain, involving large continuous output spaces over video sequences. This *value-directed* structure learning is in contrast to more data dependent structure learning, in which the structure is determined solely based upon the statistical distribution of the data, and the complexity of the model. For example, many structure learning algorithms use some simplicity prior (such as the minimum description length [Ris78,Bra99,WPG01]), and find a trade-off between the model's precision and complexity.

We now discuss a particular technique for value-directed state aggregation applied to learning the number of facial displays or gestures that need to be distinguished in our learned POMDP. As we have mentioned, the state space is represented in a factored POMDP as a product over a set of variables. In our model, the values of one of these variables, $A^{b:a}$, are the (unlabeled) gestures or facial displays. This variable splits the value function into N_a pieces, V_i, one for each value, i, of the variable $A^{b:a}$. Each such V_i gives the values of being in any state in which $A^{B:a} = i$. A similar split occurs for the policy, yielding sub-policies, π_i, giving the actions to take for each $A^{b:a} = i$. The V_i can be compared by computing the difference between them, $d_{ij} = \|V_i - V_j\|$, where $\|X\| \equiv max\{x : x \in X\}$ is the supremum norm. Two sub-policies, π_i and π_j are considered equivalent if the optimal actions agree for every state: if $\pi_i \wedge \pi_j$. These comparisons are used in the following algorithm for learning the number of display states, N_a. The algorithm starts by assigning N_a to be as large as the training data will support, and prunes redundant states.

```
repeat
  1.learn the POMDP model
  2.compute V_i and π_i ∀ i
  3.compute d_ij = ||V_i − V_j|| ∀ (i,j), i ≠ j
  4.if ∃(i,j)(π_i ∧ π_j)
  5.     {i,j} = arg min_{kl}(d_kl ∀{k,l} | π_k ∧ π_l)
  6.     merge states i and j
  7.     N_a ← N_a − 1
    end
until N_a stops changing
```

Figure 7: Procedure for value-directed structure and parameter learning for POMDPs

There are many potential ways to merge states at step 6, but we simply delete one of the the redundant states. Note that the algorithm could also start with $N_a = 2$ and add states until redundancies appear, but we have not experimented with this version [McC93]. The new states could be initialized randomly, or as a current state with added noise.

3.3.1 Complexity

Learning the POMDP parameters (step 1 in Figure 7) involves iterations of expectation-maximization, as implemented by the forwards-backwards algorithm (message passing in the Bayesian network). The complexity of this procedure is $O(N_d(N_x^2 + N_w^2)T)$, where N_x is the number of states in the dynamics process, N_w is the number of states in the configuration process, N_d is the number of high-level facial display states and T is the length of the entire sequence of data. The complexity of value iteration (step 2 in Figure 7) has a complexity of $O(N_s^2 N_a H)$, where N_s is the number of states in the POMDP, N_a is the number of actions, and H is the horizon. The remainder of the algorithm is $O(N_s^2)$ (steps 4-7 in Figure 7). Therefore, the complete learning procedure has a worst-case complexity of $O(N_d^2(N_x^2 + N_w^2)T + N_d N_s^2 N_a H)$. In typical problems, N_d, N_x and N_w are all quite small numbers, while N_s is very large (exponential in the number of variables in the POMDP). Therefore, even using simple POMDP solution approximations, the complexity will generally be dominated by the second term, $O(N_d N_s^2 N_a H)$. Attempting to compute optimal POMDP solutions would increase this complexity.

4 Experiments

To investigate the relationships between facial displays and other, conditioning factors, we adopt an experimental paradigm in which we observe humans playing computer games against other humans, or against computer agents. The following is an outline of the method:

1. Design game and encode it as a POMDP.
2. Gather training & test data sets. This involves a human playing the game either against another human or against a computer agent. In the latter case, the agent selects actions randomly in both training and test data sets.
3. Apply the procedure in Figure 7 to learn the parameters and structure of the POMDP model from the training data. Also compute an approximate policy of action.
4. Use the model to predict actions in the test data set. In the case of two humans playing against one another, the predictions are over the (observable) actions of one of the players, and can be compared to the actual actions taken in the test data for a performance measure. In the case of a human playing against an agent, the predictions are action selections from the policy, but since the agent was playing randomly, these cannot be compared to the agent's actions. Instead, we use the actions as if they were selected in the real game, and collect rewards based upon them. The total collected rewards are a performance indicator.

The following three sections describe this procedure applied to three simple games. The first (imitation game) only involves facial displays as actions, and so does not have a reward function. This game is used to explore the representational power of our computer vision modeling techniques. The second (robot control) involves a single human performing gestures for robot control, and shows how gestures can be modeled with our system. This second game also demonstrates our value-directed structure learning techniques. The third (card matching game) involves two humans playing a collaborative game. The facial displays are fairly simple, but the decision theory problem is much more complex than the other two games. This data is used to demonstrate how a policy can be computed based, in part, on non-verbal displays.

4.1 Imitation Game

To play the imitation game, a single player watches a computer animated face on a screen, and is told to imitate the actions of the face. The animated displays start from a neutral face, as shown in Figure 8(a), then warp to one of the 4 poses shown in Figures 8. The pose is held for roughly a second, and the face then warps back to the neutral pose where it remains for an additional second.

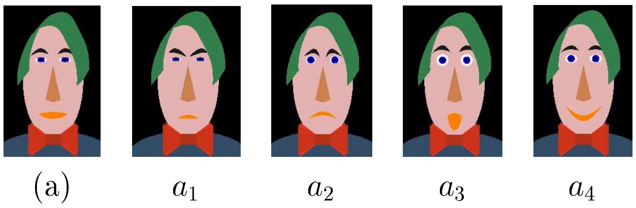


Figure 8: (a) neutral face ($C = a_1 ... a_4$) Faces which subjects were told to imitate

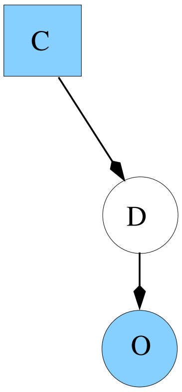


Figure 9: Time slice of graphical model of simple imitation game.

The additional independencies in this game result in a simplification of the model, as shown in Figure 9. The agent's actions, C, are choices of cartoon facial displays, $a_1 \ldots a_4$. The observations of the human's actions, $\mathbf{O}$, are sequences of video images and the spatio-temporal derivatives between subsequent video frames. The human player's actions, as modeled by the agent, are described by a discrete N_a-valued variable, $D \in \{a_1 \ldots a_{N_a}\}$, where the agent has control over N_a.

In this game, we leave out the utility function, and compute a measure of the ability of the model to represent the different imitation displays. We therefore *hide* the cartoon display labels, C, in the test data set, and compute the distribution over this variable:

$$P(C|\mathbf{O}) \propto P(\mathbf{O}|C)P(C) = \sum_i P(\mathbf{O}|D_i)P(D_i|C)$$

The maximum of this distribution tells us which cartoon display was most likely to have produced the observed human display. Agreement with the actual cartoon displays gives us an indication of how well the model can represent the display imitations.

Subjects were seated in front of a computer terminal and were told that their task is to imitate the displays on the screen. Subjects were shown each of displays initially and told to practice imitating them. Once they were satisfied with their imitations, they pressed a key, and the system began recording a video sequence through a Sony EVI-D30 color camera mounted above the computer screen. While the subjects were being recorded, the cartoon face performed a series of 40 randomly selected facial displays over a period of 2 minutes. Frames were captured at 160×120 with a BTTV frame grabber card on a desktop Pentium III PC running the Linux operating system. The frame rates were almost always above 28fps.

		cluster		
	D_1	D_2	D_3	D_4
C_1	0.01	0.01	0.61	.37
C_2	0.02	0.48	0.17	0.33
C_3	0.01	0.97	0.01	0.01
C_4	0.91	0.07	0.01	0.01

Table 1: Probability distribution $P(D|C)$ learned for subject $\mathcal{A}$

The videos from the imitation game described in the last section were temporally segmented using the onset times of the cartoon displays and the resulting sequences were input to the mixture of coupled HMM clustering and training algorithm described in Section 3 using 4 clusters (the number of displays the subjects were trying to imitate). The Viterbi algorithm was used to assign cluster membership, D, to each sequence, which were then compared to the known classes of displays the subjects were trying to imitate.

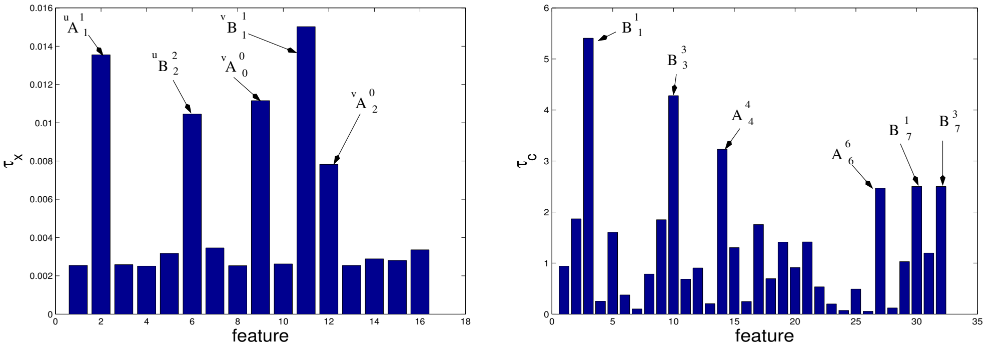
The remainder of this section evaluates the results from one of the subjects who performed the experiment. We first show the learned high-level probability distribution, $P(D|C)$, which describes the likelihood of observing each high-level motion state given each cartoon display. We then show two of the models learned for this subject: one for “smiling” imitations, and one for “surprised” imitations. For each model, we show the learned feature weights and output distributions for both dynamics and configuration chains. We also show how it analyses two sequences.

Table 1 shows the learned model parameter, $P(D|C)$, ,for subject $\mathcal{A}$, in which each row is one C state (cartoon display on screen) and each column is one recovered cluster, $D_1 ... D_4$. We see that most of the responses to the cartoon display C_4 were classified as D_1, and most of the responses to C_3 were classified as D_2. Responses to C_2 were split between those that looked the same as responses to C_3 (and so were classified as D_2, and those that looked similar to some of the responses to C_1 (classified together in state D_4. The D_3 model classified the majority of the responses to C_1. After the experiment, most subjects reported either that they did not notice a significant difference between cartoon displays a_1 and a_2, or that they could not find a way to imitate the second one, a_2, due to the extremely down-turned mouth. In the following sections, we describe each model, and show some sequences which were classified as belonging to that model.

4.1.1 Model D_1

Feature weights for the model 1 dynamics and configuration chains are shown in Figure 10. The dynamics chain has four significant feature: two in the horizontal flow components: ${}^uA_1^1, {}^vB_2^2$, and three in the vertical flow components, ${}^vA_0^0, {}^vB_1^1$, and ${}^vA_2^0$. The three most significant features in the configuration chain are B_1^1, B_3^3, and A_4^4.

The output distributions of the four states (X) in the dynamics chain are shown in Figure 11, plotted along two most significant feature dimensions, ${}^uA_1^1$ and ${}^vB_1^1$. Two states ($X = 2, 4$) correspond to no motion (the face is stationary), while the other two correspond to expansion upwards and outwards in the bottom of the face region ($X = 1$), and contraction downwards and inwards in the bottom of the face region ($X = 3$). We will see that these states correspond to the



dynamics configuration

Figure 10: Feature weights τ_k^2 for model 1 dynamics and configurations chains.

expansion and relaxation phase of smiling.

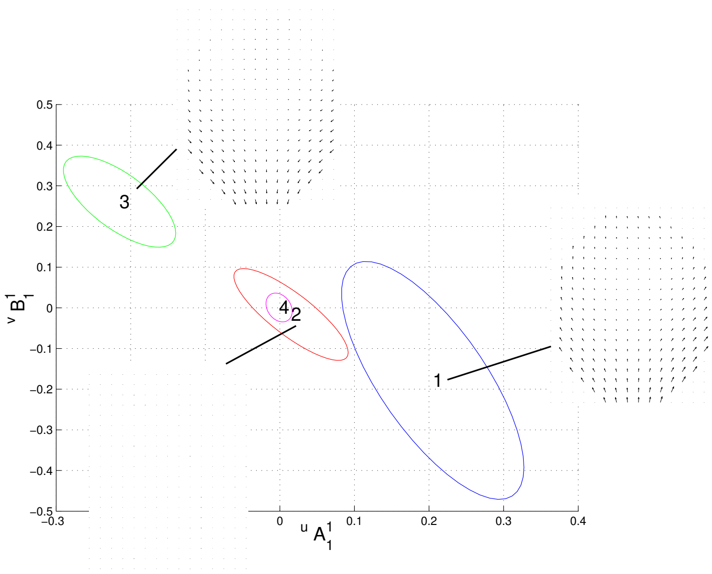


Figure 11: Dynamics chain model 1 output states plotted along two most significant dimensions according to feature weights, ${}^uA_1^1, {}^vB_1^1$. Reconstructed flow fields for X state means are also shown.

The output distributions of the four configuration states are shown in Figure 12. There are two states ($C = 2$ and $C = 4$) which describe the face in a fairly relaxed pose, while $C = 1$ and $C = 3$ describe "smiling" configurations.

Figure 2 shows model's explanation of a sequence in which $D = 1$. We see the high level distribution over D is peaked at $D = 1$. Distributions over dynamics and configuration chains show which state is most likely at each frame. The expected pose, H, and flow field, V, are shown conditioning the image, I, and the temporal derivative, f_t, respectively.

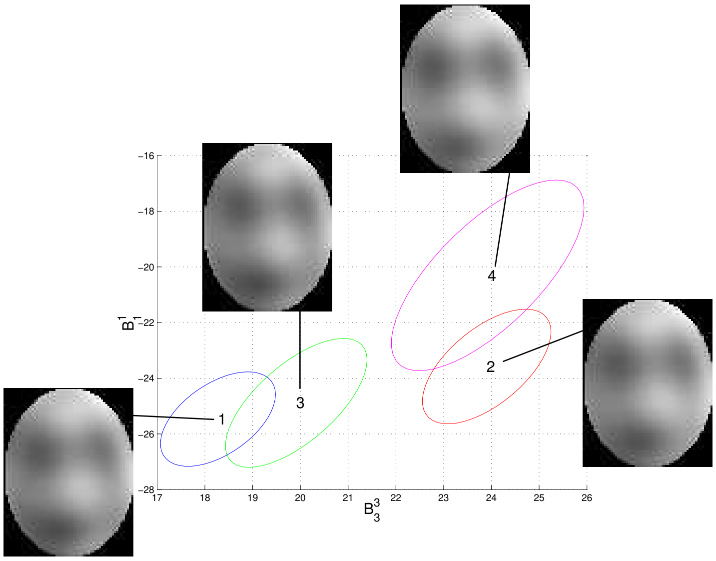


Figure 12: Configuration chain model 1 output distributions. Reconstructed grayscale images are shown for C state means.

4.1.2 Model D_2

Figure 13 shows the model's explanation of a sequence in which $D = 2$. Distributions over dynamics and configuration chains show which state is most likely at each frame. The expected pose, H, and flow field, V, are shown conditioning the image, I, and the temporal derivative, f_t, respectively. Further details and examples can be found in [Hoe04].

4.1.3 Inferring Agent Actions

We can obtain a quantitative measure of performance in this model attempting to infer the cartoon display in the test data given the human imitation. We did this analysis using a leave-one-out cross validation experiment for each of three subjects who participated in the imitation game. There were 40 sequences (imitations) for each subject, one of which was removed. The remaining 39 sequences were used to train the POMDP. The learned model was used to infer the cartoon display, C, from the remaining (left-out) sequence. The most likely value was chosen and compared to the actual display. This process was repeated three times for each subject with different random initializations, and the success rates are shown in the first row of Table 2. However, these results ignore the model's explicit representation of uncertainty, only reporting success if the actual display is the peak of $P(C|\mathbf{O})$, but in some cases, there is a second display which is nearly as likely as the best one. To demonstrate this, the second row in Table 2 shows the confusion matrices obtained if we classify the sequence correctly if it falls in the top two most likely displays, but only if the probability of the most likely display is less than 0.5. We see that many of the mis-classified sequences were assigned maximum likelihood with much uncertainty. These results can be compared to the results obtained in a supervised experiment, where each sequence is explicitly labeled, so D is observed. These results are shown in in the last row in Table 2.

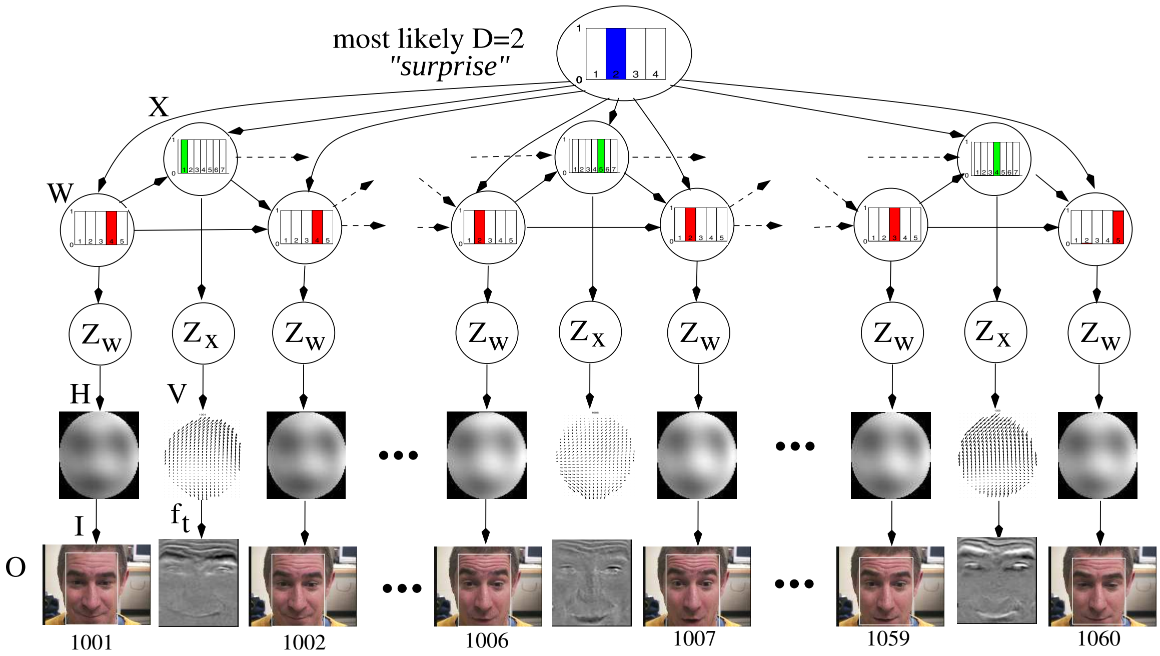


Figure 13: The face starts in its rest configuration ($W = 4$), expands with the $X = 1$ flow field towards a "eyebrows raised" ($W = 2$) configuration, and the mouth opens with the $X = 5$ flow field, resulting in a "surprised" configuration ($W = 3$). Finally, the face relaxes with the $X = 4$ flow field to a rest pose ($W = 5$).

	Subject A	Subject B	Subject C
top 1	78%	74%	62%
top 2	95%	93%	84%
supervised	82%	97%	82%

Table 2: Confusion matrices and success rates from cross-validation experiments inferring cartoon displays from sequences of three subjects. Top row: rates for match of actual display with most likely inference. Middle row: rates for match of actual display with either of the two most likely inferred displays. Bottom row: supervised results.

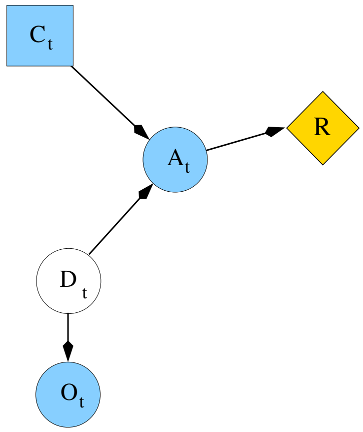


Figure 14: Two time slices of a POMDP for robot control gestures. C is the robot's action (one of *go left, go right, stop, forwards*), A is the operator's action (reward or punish the robot), R is the reward function (a one-to-one mapping from A), D is the robot's interpretation of the control command (gesture), and O is the video sequence observation of the gesture.

4.2 Robot Control Gestures

This "game" involves a human operator issuing navigation commands to a robot using hand gestures. The robot has four possible navigation actions, *go left, go right, stop* and *go forwards*, and the human operator uses four distinct hand gestures corresponding to each command. The robot, however, must learn the mapping from hand gestures to its actions. It learns this mapping during a training phase during which it acts randomly in response to the operator's hand gestures. Rewards of one unit are explicitly assigned by the operator if the robot performs the correct actions. The robot gets no reward if it performs the wrong action.

Again, each interaction in this game is temporally independent. The POMDP for each time slice is shown in Figure 14 from the point of view of the robot. The robot's action is denoted C, while the operator's action (to reward the robot or not) is explicitly represented in the model as A^b, and is fully observable. Thus, the reward function is a one-to-one mapping from this action. The robot's observations of the operator's hand gesture, O, is conditioned on the high-level interpretation of the gesture, D, which is a discrete-valued variable with N_a values.

An optimal policy of action in this model needs only be computed over a horizon of one time step (since the actions are temporally independent). The policy, π, specifies an action for each possible recognized gesture, D, such that $C = \pi(D = i)$ is the action which will most likely result in the operator rewarding the robot if $D = i$ is observed. Notice that the MDP approximation can easily lead to sub-optimal action choices. Suppose that the distribution over gesture interpretations, $P(D|\mathbf{O})$, has a roughly equal value for two values of D, so the robot is uncertain about which gesture was actually performed. The MDP approximation will simply choose the most likely one, possibly leading to an error. An optimal solution to the POMDP would include this uncertainty, and might specify some other action (such as *ask for clarification*) in the case of such uncertainty.

We recorded a set of examples of four hand gestures, designed for simple robotic direction control: *forwards*, *stop*, *go left* and *go right*. A dozen examples of each gesture were performed by a single subject in front of a stationary camera during a training session. Video was grabbed from a IEEE 1394 (Firewire) camera at 150×150 with a narrow field of view. The region of interest was taken to be the entire image, and so no tracking was required. Clearly, this would only be possible with a static camera. Sequences were taken of a fixed length of 90 frames. A robotic agent (not embodied at this stage) chose actions in response to each gesture according to a random policy, and was rewarded by the operators *good* or *bad* action, *Aact*, for choosing the correct action.

We trained the POMDP with $N_a = 6$ states. The value function and policy are shown in Figure 15 as decision diagrams. To read these diagrams, simply trace a path from the root to a leaf. The variables encountered on the nodes, and their values encountered on the edges are the state of the world, and the leaf of the value function contains the expected value of being in that state, while the leaf of the policy gives the optimal action to take in that state. Recall that $Acom \equiv D$, $Aact \equiv A$ and $Bact \equiv C$. The policies for states d_2 and d_5 are equivalent and their values are identical, and so the value-directed structure learning algorithm merges them first by simply deleting state d_5. The POMDP is re-trained, resulting in a five-state value function (not shown), in which two more states are found to agree and are merged. Again the POMDP is re-trained, this time giving a value function and policy in which no displays are found to be redundant. The final policy is a one-to-one mapping from recognised gestures ($d_1 \ldots d_4$) to actions *left, right, stop* and *forward*.

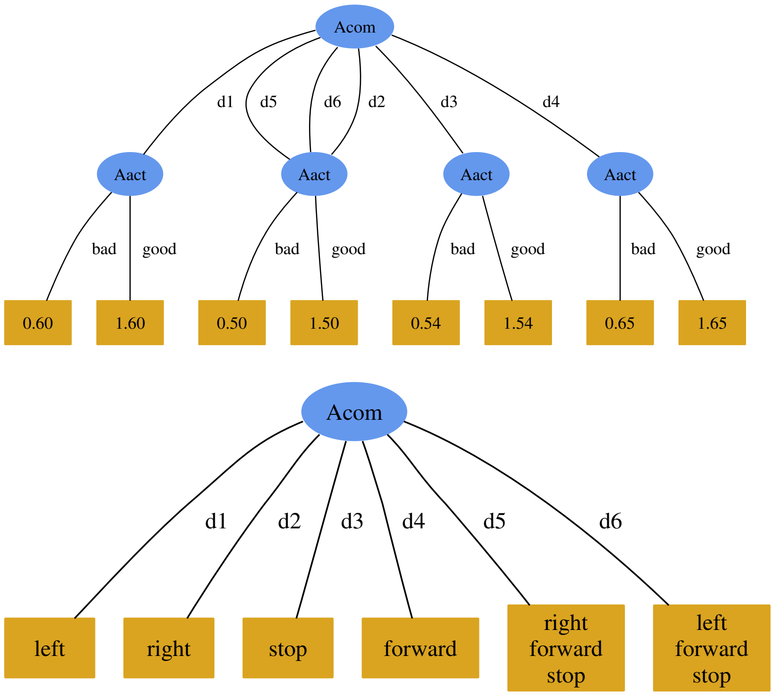


Figure 15: Original six-state value function (top) and policy (bottom), shown as decision diagrams. States are the labels on each path from the root to a leaf, which contains the value or optimal action for that state.

Figure 16 shows the model's interpretation of a part of a *stop* sequence, classified as model d_3, in the new 4-state POMDP. Stop gestures consist of an expansion phase followed by a retraction phase. left as it closes.

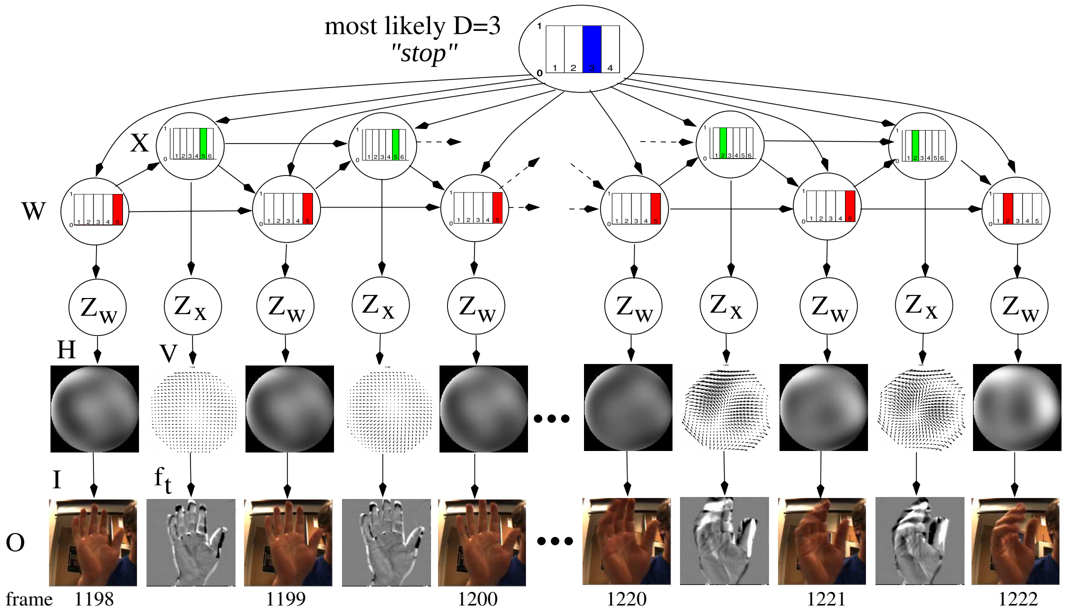


Figure 16: Final 4-state gesture model's explanation of a stop gesture as d_3.

To evaluate how well the model chooses actions, we performed a cross-validation experiment in which the POMDP was trained on all but one sequence of each gesture. The model was then used to choose actions based upon the four sequences left out. If the action is correct, one reward is given. This process is repeated for 12 different sets of four test sequences, and the total rewards gathered give an indication of how well the model performs on unseen data. The model chose the correct action 47 out of a total of $12 \times 4 = 48$ times, for a total success rate of 47/48 or 98%. The one failure was due to a mis-classification of a "left" gesture as a "right" gesture due to a large rightwards motion of the hand at the beginning of the stroke. The final POMDP models learned that there were $N_a = 4$ states in all 12 cases.

4.3 Card Matching Game

Two teams of two players play the card matching game. At the start of a round, each player is dealt three cards: a heart, a diamond and a spade. Each player can only see his own set of cards. The values of the cards (ranging from ace to ten), and their placement on the table, are randomly distributed. Each player's cards are dealt from a different deck, which is re-shuffled after every round. The players all play a single card simultaneously, and if the suits of the cards played by the two members of a team match, then that team *reserves* the sum of the values on the two cards, otherwise, that team *reserves* zero. The team with the highest reserve *wins* their reserve, while the other team wins nothing (and loses their reserve). On alternate rounds, a player has an opportunity to send a confidential *bid* to his partner, indicating a card suit. The *bids* are non-binding and do not directly affect the payoffs in the game. Finally, each player can see (but not hear) his teammate through a real-time video link. This game constrains the players to use their gestures by not having an audio link, so the players cannot speak to one another. Players cannot see members of the opposing team, nor can they see the confidential bids of the opposing team. One of the two teams is simply implemented in software, and does not actually have a video link or a bidding process, but always chooses the optimal set of matching cards.

The card matching game was played by two students in our laboratory, "Bob" and "Ann" through a computer interface. A picture of Bob's game interface during a typical interaction is shown in Figure 17. Each player viewed their partner through a direct link from their workstation

stage 1 stage 2

Figure 17: Bob's game interfaces during a typical round. His cards are face up below the "table", while Ann's cards are above it. The current bid is shown below Bob's cards, and the winnings are shown along the bottom. The cards along the sides belong to another team, which is introduced only for motivation. A bid of hearts in stage 1 is accepted by Ann, and both players commit their heart in stage 2.

to a Sony EVI s-video camera mounted about their partner's screen. The average frame rate at 320×240 resolution was over 28fps. The rules of the game were explained to the subjects, and they played four games of five rounds each. The players had no chance to discuss potential strategies before the game, but were given time to practice.

We will use data from Bob's bidding rounds in the first three games to train the POMDP model. Observations are three or four variable length video sequences, and the actions and the values of the cards of both players, as shown in Table 3. The learned model's performance will then be tested on the data from Bob's bidding rounds in the last game. It is also possible to implement a combined POMDP for both bidding and displaying rounds [Hoe04].

There are nine variables which describe the state of the game when a player has the bid. The suit of each the three cards can be one of $\heartsuit, \diamondsuit, \clubsuit$. Bob's actions, *Bact*, can be *null* (no action), or sending a confidential bid ($bid\heartsuit, bid\diamondsuit, bid\clubsuit$) or committing a card ($cmt\heartsuit, cmt\diamondsuit, cmt\clubsuit$). Ann's observed actions, *Aact*, can be *null*, or committing a card. The *Acom* variable describes Ann's communication through the video link. It is one of N_d high-level states, $D = d_1 \dots d_{N_d}$, of

Table 3: Log for the first two bidding rounds of one of the training games. A blank means the card values were the same as the previous sequence. Ann's display, *Acom*, is the most likely as classified by the final model.

	player's cards						actions			Ann's
frames	Bob			Ann			bid	Bob	Ann	Ann's
	♡	♢	♣	♡	♢	♣		*Bact*	*Aact*	*Acom*
40-150	3	4	7	2	10	7	-	bid♣	-	d_3
151-295							♣	cmt♣	cmt♣	d_1
725-827	2	5	2	7	3	8	-	bid♢	-	d_4
828-976							♢	bid♣	-	d_4
977-1048							♣	cmt♣	cmt♣	d_1

the mixture of CHMMs model described previously. Although these states have no meaning in isolation, they will obtain meaning through their interactions with the other variables in the POMDP. The other six, observable, variables in the game are more functional for the POMDP, including the values of the cards, and whether a match occurred or not. The reward function is only based upon fully observable variables, and is simply the sum of the played card values, if the suits match. The number of display states, N_d, is learned using a value-directed structure learning technique, discussed in Section 3.3.

The reward function is only based upon fully observable variables, and so can be directly written down. Recall that, in the actual game, the players only get the sum of the values on the cards they played if the card suits match, and only if the sum is greater than the other team's sum, if the other team's cards match. However, the other team only plays a motivational role in the game, and we can disregard it here by assuming the players are rewarded by getting a match, and the reward is the sum of the card values in the POMDP, regardless of the other team's play. We are assuming that the players are striving only to play matched suits which have the greatest sum, since they have no control over the other team's play. Thus, the reward function is $Acv + Bcv$ if *match* is not *null*, otherwise it is 0.

The training data set is certainly large enough to learn models of facial displays, but is quite small when it comes to learning an optimal policy. To see why, notice that, to learn an accurate POMDP, we need examples of every possible valuable state-action pair. Even if many of the 10,000 states and 6 actions in the card matching game will never be visited, the amount of training data required is still quite large. Nevertheless, under certain symmetry assumptions, it may be possible to make near-optimal decisions based only on a small training set. Equivalently, in an on-line reinforcement learning method, it may be possible to start exploiting the learned model with little exploration. For example, if we assume that the players behaviour is symmetrical under permutation of the card suits, then we can average the conditional probability distributions over all such permutations. This allows us to *fill in* more of the model without having to explicitly explore those situations. For example, suppose that a player notices that a certain facial display in response to a bid of hearts is usually followed by her partner committing hearts. If she assumes that her partner has no special preference between the heart and the diamond suits, then she may extrapolate her experiences with hearts to diamonds, and assume that the same facial display in response to a bid of diamonds will be followed by her partner committing diamonds. However, these types of arguments are dependent on the particular game being played, and generalisations require more complex models than the POMDPs considered here. We will, however, show in Section 4.3.4 how they can be used for the card game, and how it makes for better policies when training with little data.

4.3.1 Learning a POMDP

We learned the parameters of the POMDP for the bidding rounds from Bob's perspective, using the methods described in Section 3. The model was trained with four display states, which is as large as we think it possible to learn reliable models given the training set size. Two of the learned display states described sequences with little motion ("null states). The other two corresponded roughly to "nodding" and "shaking" of the head.

A slightly modified version of the structure learning algorithm described in Section 3.3 was applied. In this case, we do not compare value functions and policies for every state of the game, as they will often not agree in many places. Instead, we look at where the majority of states agree on a merge. Two states were merged, resulting in a three-state model. The two merged models, d_1 and d_2, both described "null" sequences, with little facial motion. This redundancy shows up in the value function, $V(s)$, and in the policy. The model was re-trained with only three states after merging states d_1 and d_2. The result was, as expected, one "null" state (d_1), one "nodding" state (d_2) and one "shaking" state (d_3). The remainder of this section discusses this reduced three-state model.

4.3.2 Structure of Learned Model

Figure 18 shows the learned conditional probability distribution Ann's action, *Aact*, given the current *bid* and Ann's display, *Acom*, as a decision tree. The leaves show the distribution over the possible values of *Aact*, from top to bottom: $null, cmt\heartsuit, cmt\diamondsuit, cmt\clubsuit$. We see that, if the bid is null, we expect Ann to do nothing in response. If the bid is some suit, s, and Ann's display (*Acom*) is the "nodding" display d_2, then there is a good chance that Ann will commit her card of suit s. On the other hand, if Ann's display is the "shaking" display, d_3, or the "null" display, d_1, then we expect her to do nothing (and wait for another bid from Bob).

The conditional probability distribution of Ann's display, *Acom*, at time t, given the previous and current bids, bid_{t-1}, and bid_t, respectively, are different for each of Bob's actions. This is because Ann observes Bob's bid the moment he makes it. One example, for Bobs action $bid\diamondsuit$, is shown in Figure 19. These distributions carry two important pieces of information for Bob:

1. At the beginning of a round, any bid is likely to elicit a non-null display d_2 or d_3. As shown in Figure 19, the expected distribution over $Acom = d_1, d_2, d_3$ after action $Bact = bid\diamondsuit$ if the current bid is *null* (at the beginning of a round) and the previous display was d_1 (a null display) is $0.01, 0.49, 0.49$. Thus, d_1 (null) display is not very likely, while d_2 and d_3 (nod and shake) are equally likely.

2. A "nodding" display is more likely after a "shaking" display if the bid is changed. As shown in Figure 19, the expected distribution over $Acom = d_1, d_2, d_3$ after action $bid\diamondsuit$ if the current bid is $\heartsuit$ and the previous *Acom* was d_3 is $0.004, 0.993, 0.004$: if Bob bids diamonds and sees a d_3 display (a shake), then a bid of clubs will most likely elicit a d_2 display (a nod).

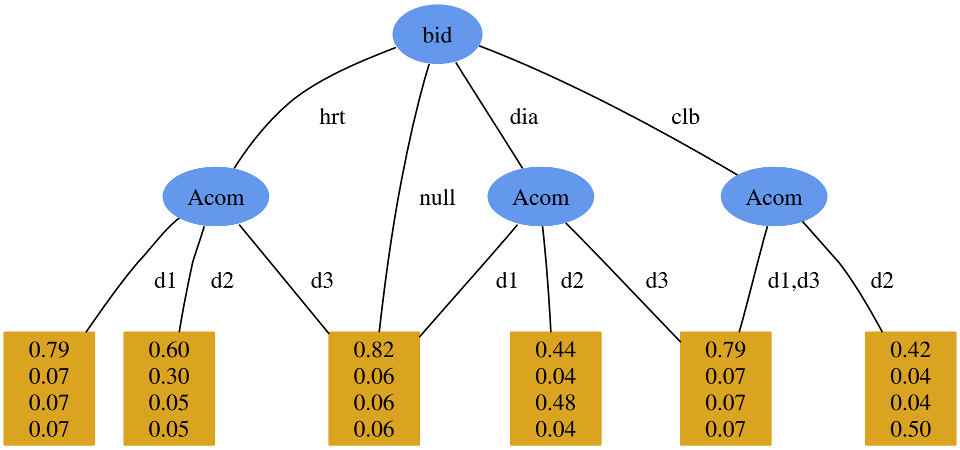


Figure 18: Learned conditional probability distribution over Ann's action, $Aact$, given the current bid and Ann's display, $Acom$. The leaves of the decision tree show the distribution over the possible values of $Aact$ from top to bottom: $null, cmt\heartsuit, cmt\diamondsuit, cmt\clubsuit$.

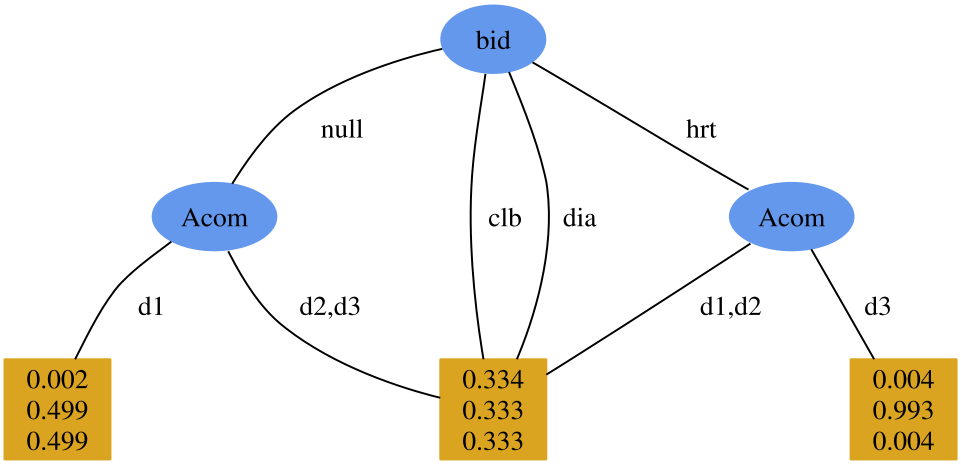


Figure 19: Learned conditional probability distribution over Ann's display, $Acom$, at time t, given the previous and current bids, bid_{t-1}, and bid_t, respectively, for Bob's action $bid\diamondsuit$. The leaves of the decision tree show the distribution over the possible values of $Acom$ from top to bottom: $d1, d2, d3$.

4.3.3 Policy of Action

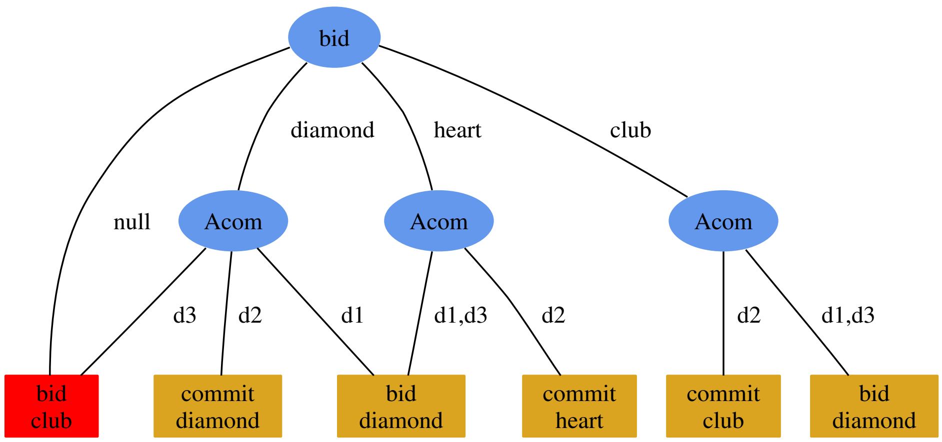


Figure 20: Policy of action in the card matching game for the situation in which $B\heartsuit v = v3, B\diamondsuit v = v3$ and $B\clubsuit v = v1$.

The full policy for the game is quite large, and we instead show the sub-policy for one particular set of values for the cards. In particular, Figure 20 shows the policy of action if the player's cards have values $B\heartsuit v = v3, B\diamondsuit v = v3$ and $B\clubsuit v = v1$. To consult the policy, simply follow the path from the root to a leaf corresponding to the state and read the recommended action at the leaf. The lightly colored leaves are actions which are "correct" in that they make sense given our interpretation of the display states (d_1 is *null*, d_2 is a nod, and d_3 is a shake). The darkly shaded nodes are actions that are "incorrect", usually due to a lack of information about these states in the training data. For example, the policy specifies that if the bid is diamonds, and the partner nodded (d_2), then commit the diamond. Otherwise, if the partner did nothing (d_1), then bid the diamond (again). Finally, if the partner shook their head (d_3), then bid the club. This last action is sub-optimal.

Table 4 show Ann's displays, the bids, Bob's cards and Bob's actions during the test game. The second-to-last column shows the predictions of Bob's actions of the policy, $\pi(s)$, discussed in the last section. This policy correctly predicted 14 out of 20 actions in the training games, and 5 out of 7 actions in the test game The incorrect actions are shaded in Table 4. However, as we have pointed out, many of the problems with this policy can be attenuated by applying symmetry arguments to construct a symmetrised policy, $\pi'(s)$. This is discussed in the next section.

4.3.4 Symmetry Considerations

As we have pointed out, symmetry arguments can be applied to the learned POMDP in order to attenuate the effects of the lack of training data. In particular, we may assume that player's do not have any particular preference over card suits, such that the conditional probability tables should be symmetric under permutation of suits. Therefore, we can "symmetrise" the probability distributions by simply averaging over the six card suit permutations. Figure 21 shows a portion of the symmetrised policy computed for the symmetrised POMDP. The predictions of this policy

game	Bob's cards			Ann's display	bid	Bob's action	policy normal	policy permute
	♡	♢	♣	*Acom*		*Bact*	$\pi(s)$	$\pi'(s)$
1	1	3	3	d_2	-	bid♢	bid♣	bid♢/♣
1				d_2	♢	cmt♢	cmt♢	cmt♢
2	2	1	3	d_2	-	bid♣	bid♣	bid♣
2				d_2	♣	cmt♣	cmt♣	cmt♣
3	3	3	1	d_2	-	bid♡	bid♣	bid♡
3				d_3	♡	bid♢	bid♢	bid♢
3				d_3	♢	cmt♢	bid♣	bid♡

Table 4: Log for the testing game showing the predicted actions from the policy, $\pi(s)$, and from the symmetrized policy, $\pi'(s)$. Shaded entries are incorrect policy predictions.

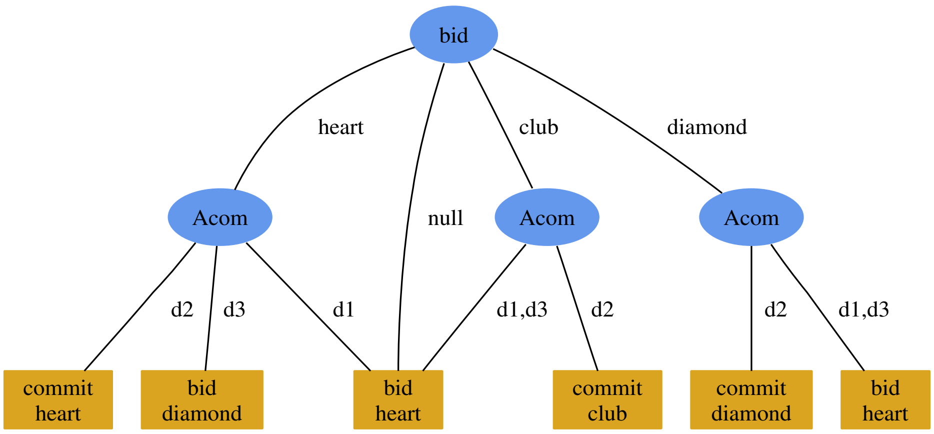


Figure 21: Policy of action in the card matching game for the situation in which $B\heartsuit v = v3$, $B\diamondsuit v = v3$ and $B\clubsuit v = v1$.

are shown in the last columns of Table 4. The symmetrised policy correctly predicts all but one of Bob's actions in the training game, for an error rate of only 5%. The mis-classification was due to the subject looking at something to one side of the screen yielding significant horizontal head motion, leading to a classificaiton as d_3.

The symmetrised policy correctly predicts all but one of Bob's actions in the test game. It does correctly predict the re-bid in the third round, where the first bid (of hearts) was refused by the partner's display. Figure 21 shows the policy of action in this situation. At the beginning, when there is no bid on the table (so *bid* is *null*), the policy recommends that Bob bid his heart (as he did). The POMDP then recognized Ann's subsequent display as state d_3 (shake), and the policy recommends changing the bid to diamonds. The POMDP classifies the following display as d_3, which is incorrect (Ann actually nodded her head), so the policy recommends bidding the heart again. The last sequence is longer than usual (over 300 frames), and includes some horizontal head motion in the beginning which appears as shaking in the model. This mis-classification may expose a weakness of the temporal segmentation method we use, which is based entirely on the observable actions and game states. Although this sequence is long, it is only the (fairly vigorous) head nod at the very end which is the important display. Perhaps a termporal segmentation which focussed more on later motions would be more attuned to this kind of sequence.

5 Conclusion

This paper has shown how partially observable Markov decision processes, or POMDPs, can be used to combine computer vision, probabilistic modeling and decision theory. The model allows an agent to incorporate actions and utilities into the sensing and representation of visual observations, and provides top-down value-based evidence for the learned probabilistic models: the agent can learn models most conducive for achieving value in a particular task. One of the key features of this technique is that it does not require labeled data sets. That is, the model makes no prior assumptions about the form or number of non-verbal behaviors used in an interaction, but rather *discovers* this from the data during training. No prior knowledge about the types of displays expected in an interaction is needed to train the model. The learned values of states are used to discover the number of display classes which are important for achieving value in the context of the interaction. This type of value-directed structure learning allows an agent to only focus resources on necessary distinctions.

This model is of interest to researchers in both computer vision and decision theory. Computer vision scientists will find a new model for human action in video streams, and a method for learning the model from data. Further, they will see how the model can be attached to a high-level decision process that implicitly defines the computer vision task. This definition is a general one: the systems must be designed so that they can learn from a set of training data. Performance can be explicitly evaluated on a task, giving the computer vision researchers solid feedback on their algorithms. Decision theorists, on the other hand, will find an output model that brings a large and important data source into contact with their models. While they have been traditionally focussing on solution techniques for "toy" problems, they are always interested in real data. The model we have presented in this paper connects them to vision data, and opens the door to research on much more difficult problem solutions than have usually been attempted.

There are a number of issues that remain to be addressed for POMDP modeling of human interactions. Perhaps the most significant is the tradeoff between spatial segmentation and spatial representation. This is closely related to the tracking problem. Is it better to track a larger number of smaller regions, with simple representations of each region, or a smaller number of larger regions, with more complex representations of each region? Our current work attempts to integrate the tracking process into the POMDP observation function, enabling the value directed learning of tracking models which are geared towards achieving value in the high-level task. In fact, it is also beleived that such learning methods will be the key to finding solutions to very large POMDPs [PB03]. We are currently working at applying our methods to robot-human interaction and robot navigation [EHL03], and to assisted living projects.

Acknowledgments. Supported by the Institute for Robotics and Intelligent Systems (IRIS), and a Precarn scholarship. We thank Pascal Poupart and the anonymous reviewers of our conference submissions for insightful comments. We thank Nicole Arksey, Don Murray, Pantelis Elinas, Andrea Bunt, Robert St-Aubin, Jochen Lang and Kasia Muldner for participating in the experiments.

References

[ABMM03] Alexei A. Afros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Proceedings of International Conference on Computer Vision*, pages 726–733, Nice, France, 2003.

[ASKP03] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering clusters in motion time-series data. In *Proceedings of Intl. Conference on Computer Vision and Pattern Recognition 2003*, pages 682–688, Madison, Wisconsin, June 2003.

[Bar01] Marian Stewart Bartlett. *Face Image Analysis by Unsupervised Learning*. Kluwer Academic Publishers, Norwell, MA, 2001.

[BD01] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(3), March 2001.

[BDH99] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artifical Intelligence Research*, 11:1–94, 1999.

[BF96] Y. Bengio and P. Frasconi. Input-output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, September 1996.

[BHK97] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, July 1997.

[BLB⁺03] M.S. Bartlett, G. Littlewort, B. Braathen, T.J. Sejnowski, and J.R. Movellan. A prototype for automatic recognition of spontaneous facial actions. In S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 382–386. MIT Press, 2003.

[BOP97] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden Markov models for complex action recognition. In *International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.

[Bra99] Matthew Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11:1155–1182, 1999.

[Bre97] Chris Bregler. Learning and recognising human dynamics in video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[BS99] Cynthia Breazeal and Brian Scassellati. A context-dependent attention system for a social robot. In *Proceedints of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99)*, pages 1146–1151, Stockholm, Sweden, 1999.

[BSA91] S.O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with moment invariants: A comparative study and new results. *Pattern Recognition*, 24(12):1117–1138, 1991.

[BY97] Michael Black and Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. *International Journal of Computer Vision*, 25(1):23–48, 1997.

[CBC⁺00] Justine Cassell, Timothy Bickmore, Lee Campbell, Hannes Vilhjámsson, and Hao Yan. Human conversation as a system framework: Designing embodied conversational agents. In Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors, *Embodied Conversational Agents*, chapter 2, pages 29–63. MIT Press, 2000.

[CdFGT03] P. Carbonetto, N. de Freitas, P. Gustafson, and N. Thompson. Bayesian feature weighting for unsupervised learning with application to object recognition. In C M Bishop and B J Frey, editors, *Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, January 2003.

[Cho91] Nicole Chovil. Social determinants of facial displays. *Journal of Nonverbal Behavior*, 15(3):141–154, Fall 1991.

[CP99] Brian Clarkson and Alex Pentland. Unsupervised clustering of ambulatory audio and video. In *Proc. ICASSP*, 1999.

[CSG⁺03] Ira Cohen, Nice Sebe, Ashutosh Garg, Lawrence S. Chen, and Thomas S. Huang. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding*, 91:160–187, 2003.

[CSPC00] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. *Embodied Conversational Agents.* MIT Press, 2000.

[CT98] Ross Cutler and Matthew Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Proc. Intl. Conference on Automatic Face and Gesture Recognition*, pages 98–104, Nara, Japan, April 1998.

[DBH$^+$99] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(10):974–989, October 1999.

[DEP96] Trevor J. Darrell, Irfan A. Essa, and Alex P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1236–1242, 1996.

[DNR77] A.P. Dempster, N.M.Laird, and D.B. Rubin. Maximum likelihood from incomplete data using the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38, 1977.

[DP96] Trevor Darrell and Alex Pentland. Active gesture recognition using partially observable Markov decision processes. In *13th IEEE Intl. Conference on Pattern Recognition*, Vienna, Austria, 1996.

[EHL$^+$02] Pantelis Elinas, Jesse Hoey, Darrell Lahey, Jeff Montgomery, Don Murray, Stephen Se, and James J. Little. Waiting with José, a vision based mobile robot. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 678–682, Washington, D.C., May 2002.

[EHL03] Pantelis Elinas, Jesse Hoey, and James J. Little. Homer: Human oriented messenger robot. In *Proc. AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, Stanford, CA, March 2003.

[EP97] I.A. Essa and A. Pentland. Coding analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):757–763, July 1997.

[EW78] Paul Ekman and W.Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, CA, 1978.

[FBYJ00] David J. Fleet, Michael J. Black, Yaser Yacoob, and Allan D. Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):171–193, 2000.

[FND03] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, March 2003.

[Fri94] Alan J. Fridlund. *Human facial expression: an evolutionary view.* Academic Press, San Diego, CA, 1994.

[FST98] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical Hidden Markov Model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[GJH99] Aphrodite Galata, Neil Johnson, and David Hogg. Learning structured behaviour models using variable length Markov models. In *IEEE Workshop on Modelling People*, Corfu, Greece, September 1999.

[Gmy02] Piotr J. Gmytrasiewicz. Toward optimal planning in multiagent environments: Basic framework. Technical report, University of Chicago, Chicago, IL, November 2002.

[HL00] Jesse Hoey and James J. Little. Representation and recognition of complex human motion. In *Proc. IEEE CVPR*, pages 752–759, Hilton Head, SC, June 2000.

[HL03] Jesse Hoey and James J. Little. Bayesian clustering of optical flow fields. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1086–1093, Nice, France, October 2003.

[Hoe01] Jesse Hoey. Hierarchical unsupervised learning of facial expression categories. In *Proc. ICCV Workshop on Detection and Recognition of Events in Video*, pages 99–106, Vancouver, Canada, July 2001.

[Hoe04] Jesse Hoey. *Decision Theoretic Learning of Human Facial Displays and Gestures.* PhD thesis, University of British Columbia, 2004.

[HS81] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HSJ95] E. Hunter, J. Schlenzig, and R. Jain. Posture estimation in reduced-model gesture input systems. In *International Workshop on Automatic Face- and Gesture-Recognition*, pages 290–295, Zurich, Switzerland, 1995.

[JP98] A. Jebara and A. Pentland. Action reaction learning: Analysis and synthesis of human behaviour. In *IEEE Workshop on The Interpretation of Visual Motion*, 1998.

[Kje01] Rick Kjeldsen. Head gestures for computer control. In *Proceedings of the Workshop on Recognition And Tracking of Face and Gesture – Real Time Systems (RATFG-RTS)*, pages 61–67, Vancouver, Canada, July 2001.

[KLC98] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[Koh89] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1989.

[LF98] Michael E. Leventon and William T. Freeman. Bayesian estimation of 3-d human motion from and image sequence. Technical Report TR-98-06, Mitsubishi Electric Research Laboratory, July 1998.

[LKCL98] James Jenn-Jier Lien, Takeo Kanade, Jeffrey F. Cohn, and Ching-Chung Li. Subtly different facial expression recognition and expression intensity estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 443–449, Santa Barbara, CA, 1998.

[LP98] Simon X. Liao and Miroslaw Pawlak. On the accuracy of Zernike moments for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(12):1358–1364, December 1998.

[LSR+00] Bastian Leibe, Thad Starner, William Ribarsky, Zachary Wartell, David Krum, Justin Weeks, Brad Singletary, and Larry Hodges. Towards spontaneous interaction with the perceptive workbench, a semi-immersive virtual environment. In *IEEE Virtual Reality*, pages 13–20, New Brunswick, NJ, March 2000.

[LTC97] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):743–756, July 1997.

[McC93] R. Andrew McCallum. Overcoming incomplete perception with utile distinction memory. In *Proc. Tenth International Machine Learning Conference*, 1993.

[McN92] David McNeill. *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, Chicago, IL, 1992.

[MP91] Kenji Mase and Alex Pentland. Recognition of facial expression from optical flow. *IEICE Transactions E*, 74(10):3474–3483, 1991.

[MP01] Kevin Murphy and Mark Paskin. Linear time inference in hierarchical HMMs. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, Vancouver, BC, 2001.

[MPR+02] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[Mur02] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, UC Berkeley, Computer Science Division, July 2002.

[MYD96] Carlos Morimoto, Yaser Yacoob, and Larry Davis. Recognition of head gestures using hidden Markov models. In *Proceeding of ICPR*, pages 461–465, Austria, 1996.

[Mye91] Roger B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, Cambridge, Massachussetts, 1991.

[OHG02] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered representations for human activity recognition. In *Proceedings of International Conference on Multimodal Interfaces*, Pittsburgh, PA, October 2002.

[PB03] Pascal Poupart and Craig Boutilier. Value-directed compression of POMDPs. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pages 1547–1554, Vancouver, 2003.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[Pen00] Alex Pentland. Looking at people: sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119, 2000.

[PFH99] Vladimir Pavlovic, Brendan J. Frey, and Thomas S. Huang. Variational learning in mixed-state dynamic graphical models. In *Proc. Uncertainty in Artificial Intelligence (UAI)*, pages 522–530, Stockholm, Sweden, July 1999. Morgan Kaufmann.

[PH00] Tim Paek and Eric Horvitz. Conversation as action under uncertainty. In *Proceedings of Uncertainty in Artificial Intelligence*, Stanford, CA, June 2000.

[PR89] Aluizio Prata and W.V.T. Rusch. Algorithm for computation of Zernike polynomials expansion coefficients. *Applied Optics*, 28(4):749–754, February 1989.

[RFD97] James A. Russell and Jose Miguel Fernández-Dols. What does facial expression mean? In James A. Russell and Jose Miguel Fernández-Dols, editors, *The Psychology of Facial Expression*, chapter 1, pages 3–30. Cambridge University Press, Cambridge, UK, 1997.

[Ris78] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[RPT00] N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000)*, Hong Kong, 2000.

[SAH91] E.P. Simoncelli, E.H. Adelson, and D.J. Heeger. Probability distributions of optical flow. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 310–315, Mauii, Hawaii, USA, 1991.

[SHJ94] J. Schlenzig, E. Hunter, and R. Jain. Vision based hand gesture interpretation using recursive estimation. In *Proc. Asilomar Conference on Signals, Systems and Computation*, pages 394–399, October 1994.

[Smy97] Padhraic Smyth. Clustering sequences with hidden Markov models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 10, 1997.

[SP95] Thad Starner and Alex P. Pentland. Visual recognition of american sign language using hidden Markov models. In *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, Zurich, Switzerland, 1995.

[TC88] Cho-Huak Teh and Roland T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, July 1988.

[Tea80] Michael Reed Teague. Image analysis via the general theory of moments. *Journal of Optical Society of America*, 70(8):920–930, 1980.

[TK93] D. Terzopoulus and K.Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(6):569–579, June 1993.

[TKC01] Yingli Tian, Takeo Kanade, and Jeffrey F. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), February 2001.

[TP91] Matthew Turk and Alex P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[VM98] Christian Vogler and Dimitris Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *IEEE Intl. Conference on Computer Vision*, pages 363–369, Mumbai, India, 1998.

[vZ34] F. von Zernike. Beugungstheorie des schneidenvarfahrens und seiner verbesserten form, der pahsekontrastmethode. *Physica*, I:689–704, 1934.

[WPG01] Michael Walter, Alexandra Psarrou, and Shaogang Gong. Data driven gesture model acquisition using minimum description length. In *Proc. British Machine Vision Conference*, Manchester, UK, September 2001.

[ZCMG01] Bo Zhang, Qinsheng Cai, Jianfeng Mao, and Baining Guo. Planning and acting under uncertainty: A new model for spoken dialogue system. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 572–579, Seattle, WA, August 2001.

[ZL02] Dengsheng Zhang and Guojun Lu. An integrated approach to shape based image retrieval. In *Proceedings of 5th Asian Conference on Computer Vision (ACCV)*, Melbourne, Australia, January 2002.