Layered Representations for Learning and Inferring Office Activity from Multiple Sensory Channels

Nuria Oliver[a,*] Ashutosh Garg[b] Eric Horvitz[a]

[a] *Adaptive Systems & Interaction, Microsoft Research, Redmond, WA*

[b] *Comp. Science Dept., Univ. Illinois Urbana-Champaign, Champaign, IL*

Corresponding author

Nuria Oliver

Adaptive Systems & Interaction, Microsoft Research, Redmond, WA 98052

Phone: +1 425 705 4051

Fax: +1 425 936 7329

Email: nuria@microsoft.com

[*] Corresponding author.
Email addresses: nuria@microsoft.com (Nuria Oliver), ashutosh@uiuc.edu (Ashutosh Garg), horvitz@microsoft.com (Eric Horvitz).
URL: http://research.microsoft.com/∼nuria (Nuria Oliver).

Preprint submitted to Elsevier Science *19 February 2004*

Abstract

We present the use of layered probabilistic representations for modeling human activities, and describe how we use the representation to do sensing, learning, and inference at multiple levels of temporal granularity and abstraction and from heterogeneous data sources. The approach centers on the use of a cascade of Hidden Markov Models named Layered Hidden Markov Models (LHMMs) to diagnose states of a user's activity based on real-time streams of evidence from video, audio and computer (keyboard and mouse) interactions. We couple these LHMMs with an expected utility analysis that considers the cost of misclassification. We describe the representation, present an implementation, and report on experiments with our layered architecture in a real-time office-awareness setting.

Key words: Office Awareness; Office activity recognition; Multi-modal systems; Human Behavior Understanding; Hidden Markov Models

Preprint submitted to Elsevier Science *19 February 2004*

List of symbols

T_L, T_{L+1}

$O^L = \{O^L_1, O^L_2, ..., O^L_{T_L}\} = O^L(1:T_L)$

K_L

$\mathcal{X}^{T_L}$

O^L_i

$O^L \in \mathcal{X}^{T_L}$

$f_L : \mathcal{X}^{T_L} \to \mathcal{Y}^L$

$\mathcal{Y}^L \in \{1, ..., K_L\}$

f_L

K, L, $L+1$

$\mathcal{X}^{T_{L+1}} = \{\mathcal{Y}^L_1, ..., \mathcal{Y}^L_{T_{L+1}}\}$

$f_{L+1} : \mathcal{X}^{T_{L+1}} \to \mathcal{Y}^{L+1}$

$\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i) = P(q_t = S_i|O_1...O_t)$

S_i, t

$\{O_1...O_t\}$, $O(1:t)$

$\mathcal{L} = P(O_1...O_T) = \sum_{i=1}^N \alpha_t(i)\beta_t(i)$

N

$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i)P_{j|i}]p_j(o_t)$

$\beta_t(i) = [\sum_{j=1}^N \beta_{t+1}(j)P_{i|j}p_j(o_{t+1})]$

$P_{i|j}$

$p_i(o_t)$, i, j, α, β

M^L_k, $k = 1, ..., K$

$\mathcal{L}(k)^L_t = \log(P(O(1:t)|M^L_k)) = \log\sum_i \alpha_t(i;M^L_k)$

$\alpha_t(i;M^L_k)$

$\alpha_{t+1}(j;M^L_k) = \sum_{i=1}^N \alpha_t(i;M^L_k)P^{M^L_k}_{j|i}p_j(o_t;M^L_k)$

$P^{M^L_k}_{j|i}$

$p_j(o_t;M^L_k)$, o_t

$C(t)^L = \arg\max_k \mathcal{L}(k)^L_t$

τ

$C(1:\tau)^L$

$\{\mathcal{L}(1:K)^L_{t=1}, ..., \mathcal{L}(1:K)^L_{t=\tau}\}$

$Zero_s(m) = \frac{1}{N}\sum_{n=m-N+1}^m \frac{|sign(s(n))-sign(s(n-1))|}{2}\cdot w(m-n)$

m, N, w, $s(n)$, $x_i(n)$

$sign(s(n)) = \{+1,\ s(n)\geq 0; -1,\ s(n)<0\}$

$x_i(n) = a_i s(n-t_i) + b_i(n)$

$x_i(n) = g_i * s(n-t_i) + b_i(n)$

a_i, b_i, g_i

$[N_1, N_2]$

$r_{lr}(d) = \sum_{n=N_1}^{N_2} l(n)r(n-d)$

$Norm\mathcal{L}_i = \frac{\mathcal{L}_i - min_j(\mathcal{L}_j)}{max_j(\mathcal{L}_j)-min_j(\mathcal{L}_j)}$

$i = 1, ..., N$, $j = 1, ...N$

1 Introduction

Researchers and application developers have long been interested in the promise of performing automatic and semi-automatic recognition of human behavior from observations. Successful recognition of human behavior is critical in a number of compelling applications, including automated visual surveillance and multimodal human–computer interaction (HCI)—user interfaces that consider multiple streams of information about a user's behavior and the context of a situation. Context is a key element in the interaction between humans and computers, describing the surrounding facts that add meaning. Although progress has been occurring on multiple fronts, many challenges remain for developing machinery that can provide rich, human-centric notions of context. By improving the computer's access to context, we can increase the richness of the communication between humans and computers and catalyze the development of new kinds of computational services and experiences.

We describe in this paper our efforts to build probabilistic machinery that can provide real-time interpretations of human activity in and around an office. The paper is organized as follows: We first provide background on context-sensitive systems and representations related to the one proposed in this paper in Section 2. Section 3 describes the challenge of understanding human activity in an office setting, reviews the kinds of perceptual inputs we wish to analyze, and the problems incurred with a single-layer (non-hierarchical) implementation of HMMs. In Section 4, we introduce our representation, based on Layered Hidden Markov Models (LHMMs). Section 5 presents the architecture and implementation of a system named SEER that uses LHMMs, and describes the details of feature extraction, learning and classification used in the system. Experimental results with the use of SEER are reviewed in Section 6. Finally, we summarize our work and highlight several future research directions in Section 7.

2 Previous Work

2.1 Context Sensitive Systems

Location and identity have been the most common properties considered as comprising a user's context in "context-aware" HCI systems. Context can include other critical aspects of a user's situation, such as the user's current and past activities and intentions. Recent work on probabilistic models for reasoning about a user's location, intentions, and focus of attention have highlighted opportunities for building new kinds of applications and services [1,2].

Most of the previous work on leveraging perceptual information to recognize human activities has centered on the identification of a specific type of activity in a particular scenario. Many of these techniques are targeted at recognizing single, simple events, *e.g.,* "waving the hand" or "sitting on a chair". Less effort has been applied to research on methods for identifying more complex patterns of human behavior, extending over longer periods of time and involving multiple people. Dynamic models of periodic patterns of people's movements are used by Davis et al. [3] to model the periodicity of activities such as walking. Other approaches to the recognition of human activity employ graphical models. A significant portion of work in this arena has harnessed Hidden Markov Models (HMMs) [4]. Starner and Pentland in [5] use HMMs for recognizing hand movements used to relay symbols in American Sign Language. The different signs are recognized by computing the probabilities that models for different symbols would have produced the observed visual sequence. More complex models, such as Parameterized-HMMs [6], Entropic-HMMs [7], Variable-length HMMs [8] and Coupled-HMMs [9,10], have been used to recognize more complex activities such as the interaction between two people. Bobick and Ivanov [11], propose the use of a stochastic context-free grammar to compute the probability of a temporally consistent sequence of primitive actions recognized by HMMs. Clarkson and Pentland model events and scenes from audiovisual information in [12]. They have developed a wearable computer system that automatically clusters audio-visual information into events and scenes in a hierarchical manner. Their goal is to determine where the user is at each instant of time (*i.e.* at home, the office, at the bank, etc). Brand and Kettnaker in [7] propose an entropic-HMM approach to segment the observed video activities (office activity and outdoor traffic) into meaningful states. They illustrate their models in video monitoring of office activity and outdoor traffic. In [13], a probabilistic finite-state automaton (a variation of structured HMMs) is used for recognizing different scenarios, such as monitoring pedestrians or cars on a freeway.

Although HMMs appear to be robust to changes in the temporal segmentation of observations, they suffer from a lack of structure, an excess of parameters, and an associated over-fitting of data when they are applied to reason about long and complex temporal sequences with insufficient training data. Finally, in recent years, more complex Bayesian networks have also been adopted for the modeling and recognition of human activities [14,15,2,16–21].

To date, however, there has been little research on real-time, multimodal systems for human-computer interaction that use statistical methods to model typical human activities in a hierarchical manner and for long periods of time.

We have developed a probabilistic representation based on a tiered formulation of dynamic graphical models that we refer to as Layered Hidden Markov Models (LHMMs). The methods and working system described in this paper focus on this representation. We show how with our approach one can learn and recognize on the fly common situations in office settings. Before describing the details of our approach, we shall compare the proposed architecure of LHMMS with related probabilistic representations.

2.2 Related Representations

HMMs and their extensions, including the architecture proposed in this paper (LHMMs), are particular cases of temporal or dynamic graphical models (DGMs). DGMs consist of a set of random variables represented as nodes as well as directed edges or links between them. They define a mathematical form of the joint or conditional Probability Distribution Function (PDF) between the random variables. DGMs provide a probabilistic, graphical representation of conditional dependencies (causality) among variables. Therefore, given suitable independence relationships among the variables over time, DGMs can provide a computationally efficient and sufficiently expressive solution to the problem of human activity modeling and recognition. The absence of directed links between nodes implies a conditional independence. Moreover a family of transformations can be performed on the graphical structure that has a direct translation in terms of mathematical operations applied to the underlying PDF. Finally they are modular, *i.e.,* one can express the joint global PDF as the product of local conditional PDFs.

DGMs present several important advantages that are relevant to the problem of human behavior modeling: they can handle incomplete data as well as uncertainty; they are trainable; they encode causality (conditional independency) in a natural way; algorithms exist for doing predictive inference; they offer a framework for combining prior knowledge and data; and finally they are modular and parallelizable.

A layered structure, in addition, provides several valuable properties. A layered formulation makes it feasible to decouple different levels of analysis for training and inference. As it is further explained in Section 4.1, each level of our hierarchy is trained independently, with different feature vectors and time granularities. Once the system has been trained, inference can be carried out at any level of the hierarchy. One could retrain the lowest (most sensitive to variations in the environment) level, for example, without having to retrain any other level in the hierarchy.

A formulation for Hierarchical HMMs (HHMMs) was first proposed in [28] in work that extended the standard Baum-Welch procedure and presented an estimation procedure of the model parameters from unlabeled data. A trained model was applied to an automatic hierarchical parsing of an observation sequence as a dendrogram. Because of the computational complexity of the original algorithm, the authors suggest an efficient approximation to the full estimation scheme. The approximation could further be used to construct models that adapt both their topology and parameters. The authors briefly illustrate the performance of their models on natural written English text interpretation and in English handwriting recognition. Recently, Murphy and Paskin introduce in [29] a linear-time inference algorithm for HHMMs.

In [21], Hoey proposes the use of a hierarchical framework for event detection. Although being a nice framework it does not seem to be particularly suited for a task with real-time constraints, because it requires the manual segmentation of the audio/video streams. A new architecture of HMMs called *embedded HMMs* is proposed in [30]. Such embedded HMMs are used in applications that handle two-dimensional data such as images. One HMM models one dimension of the data while its state variables correspond to the other dimension of the data. They have successfully applied these to the task of face recognition.

In the original formulation of [28] and other related papers ([21,29]), each state of the architecture is another HMM or variation, and therefore represents a time sequence of the raw signals. In our model, however, at any given level of the hierarchy, there are multiple HMMs each corresponding to a certain concept (for example, we have five HMMs corresponding to different classes of audio signals - speech, silence, music, etc). These HMMs take as observations either the features computed from the raw signals –at the lowest level– or the inferential results from the previous level –at any other level.

The LHMM approach is most closely related to the concept of Stacked Generalization [31], where the main idea is to learn classifiers on top of classifiers. Stacked Generalization is a technique proposed to use learning at multiple levels. A learning algorithm is used to determine how the outputs of the base classifiers should be combined. For example, in a two-layer stacked classifier, the original dataset constitues the "level zero" data. All the base classifiers are run at this level. The "level one" data are the outputs of the base classifiers. Another learning process occurs using as input the "level one" data and as output the final classification results. This is a more sophisticated technique than cross-validation and has been shown to reduce the classification error due to the bias in the classifiers. Note that, while HMMs are generative probabilistic models, they can also be treated as classifiers.

From this perspective, we can describe LHMMs as a representation for learning different stacked classifiers and using them to do the classification of temporal concepts with different time granularities. Rather than training the models at all the levels at the same time, the parameters of the HMMs at each level can be trained independently –provided that the previous level has been already trained, in a bottom-up fashion. The inputs (observations) of each level are the classification outputs of the previous level, such that only at the lowest level the observations (the leaves of the tree) are the feature vectors extracted directly from sensor signals.

3 Tractable and Robust Context Sensing

A key challenge in inferring human-centric notions of context from multiple sensors is the fusion of low-level streams of raw sensor data –for example, acoustic and visual cues– into higher-level assessments of activity. The task of moving from low-level signals to more abstract hypotheses about activity brings into focus a consideration of a spectrum of approaches. Potentially valuable methods include template matching, context-free grammars, and various statistical methods. We have developed a probabilistic representation based on a tiered formulation of dynamic graphical models that we refer to as Layered Hidden Markov Models (LHMMs).

To be concrete, we have explored the challenge of fusing information from the following sensors:

1. Binaural microphones: Two mini-microphones ($20 - 16000$ Hz, SNR 58 dB) capture ambient audio information and are used for sound classification and localization. The audio signal is sampled at 44100 KHz.

2. Camera: A video signal is obtained via a standard Firewire camera, sampled at 30 f.p.s, and it is used to determine the number of persons present in the scene.

3. **Keyboard and mouse:** We keep a history of keyboard and mouse activities during the past 1, 5 and 60 seconds.

Initially, we built single-layer (non-hierarchical) models to reason about the overall office situation, including determining the presence of a PHONE CONVERSATION, A FACE TO FACE CONVERSATION, A ONGOING PRESENTATION, A DISTANT CONVERSATION, NOBODY IN THE OFFICE and A USER IS PRESENT AND ENGAGED IN SOME OTHER ACTIVITY. Some of these activities have been proposed in the past as indicators of a person's availability [22]. We explored the use of Hidden Markov Models (HMMs), a popular probabilistic framework for modeling processes that have structure in time. An HMM is essentially a quantization of a system's configuration space into a small number of discrete states, together with probabilities for the transitions between states. A single finite discrete variable indexes the current state of the system. Any information about the history of the process needed for future inferences must be reflected in the current value of this state variable. There are efficient algorithms for state and parameter estimation in HMMs. Graphically HMMs are often depicted "rolled-out in time", such as in Figure 1 (a).

We found, however, that a single-layer HMM approach generated a large parameter space, requiring substantial amounts of training data for a particular office or user, and with typical classification accuracies not high enough for a real application. Finally and more importantly, when the system was moved to a new office, copious retraining was typically necessary to adapt the model to the specifics of the signals and/or user in the new setting.

Therefore, we sought a representation that would be robust to typical variations within office environments, such as changes of lighting and acoustics, and that would allow the models to perform well when transferred to new office spaces with minimal tuning through retraining. We also pursued a representation that would map naturally onto the problem space. Psychologists have, in fact, found that many human behaviors are hierarchically structured [23]. We converged on the use of a multilevel representation that allows for explanations at multiple temporal granularities, by capturing different levels of temporal detail.

4 LHMMs: Layered Hidden Markov Models

We have developed a Layered HMM (LHMM) representation in an attempt to decompose the parameter space in a way that could enhance the robustness of the system by reducing training and tuning requirements. In LHMMs, each layer of the architecture is connected to the next layer via its inferential results. The representation segments the problem into distinct layers that operate at different temporal granularities [1] —allowing for temporal abstractions from pointwise observations at particular times into explanations over varying temporal intervals. LHMMs can be regarded as a cascade of HMMs. The structure of a three-layer LHMM is displayed in Figure 1 (b).

Formally, given a set of T_L observations, $O^L = \{O^L_1, O^L_2, ..., O^L_{T_L}\} = O^L(1 : T_L)$, at level L, we can consider the HMMs at this level as being a multiclass classifier mapping these T_L observations to one of K_L classes. In our case, each HMM at each level of the hierarchy models one class. Let $\mathcal{X}^{T_L}$ be the

[1] The concept of "time granularity" in this context corresponds to the window size or vector length of the observation sequences in the HMMs.

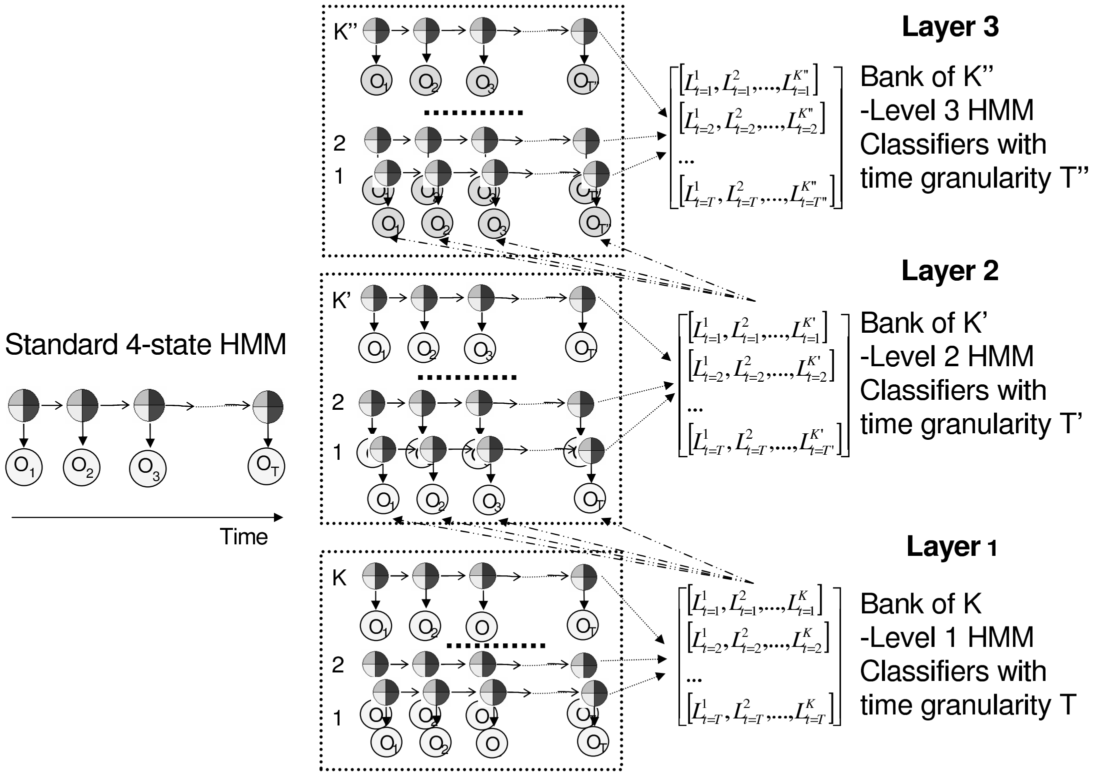

Standard 4-state HMM
O_1 O_2 O_3 O_T
Time
K''
2
1
K'
K
Layer 3
$[L^1_{t=1}, L^2_{t=1}, ..., L^{K''}_{t=1}]$
$[L^1_{t=2}, L^2_{t=2}, ..., L^{K''}_{t=2}]$
...
$[L^1_{t=T}, L^2_{t=T}, ..., L^{K''}_{t=T''}]$
Bank of K'' -Level 3 HMM Classifiers with time granularity T''
Layer 2
$[L^1_{t=1}, L^2_{t=1}, ..., L^{K'}_{t=1}]$
$[L^1_{t=2}, L^2_{t=2}, ..., L^{K'}_{t=2}]$
...
$[L^1_{t=T}, L^2_{t=T}, ..., L^{K'}_{t=T'}]$
Bank of K' -Level 2 HMM Classifiers with time granularity T'
Layer 1
$[L^1_{t=1}, L^2_{t=1}, ..., L^{K}_{t=1}]$
$[L^1_{t=2}, L^2_{t=2}, ..., L^{K}_{t=2}]$
...
$[L^1_{t=T}, L^2_{t=T}, ..., L^{K}_{t=T}]$
Bank of K -Level 1 HMM Classifiers with time granularity T

Fig. 1. Graphical Representation of (a) HMMs and rolled-out in time, and (b) an architecture of Layered HMMs with 3 different levels of temporal granularity

sample space of vectors O_i^L. If $O^L \in \mathcal{X}^{T_L}$, then the bank of K_L HMMs[2] can be represented as $f_L : \mathcal{X}^{T_L} \to \mathcal{Y}^L$, where $\mathcal{Y}^L \in \{1, ..., K_L\}$ is the discrete variable with the class label. *i.e.* the bank of HMMs is a function f_L that outputs one class label every T_L observations. The HMMs at the next level $(L+1)$ take as inputs the outputs of the HMMs at level L, *i.e.* $\mathcal{X}^{T_{L+1}} = \{\mathcal{Y}_1^L, ..., \mathcal{Y}_{T_{L+1}}^L\}$, and learn a new classification function with time granularity T_{L+1}, $f_{L+1} : \mathcal{X}^{T_{L+1}} \to \mathcal{Y}^{L+1}$.

4.1 Learning in LHMMs

In this framework, each layer of HMMs is learned independently of the others, with one HMM per class. The availability of labeled data during the training phase allows us to do efficient supervised learning. By itself, each layer is trained using the same learning and inference machinery that it is used for HMMs.

The problem of learning the model parameters in an HMM is solved by the

[2] Note that we have one HMM for each class. We will denote these kind of HMMs *discriminative* HMMs.

forward-backward or Baum-Welch algorithm. This algorithm provides expressions for the forward, $\alpha_t(i)$, and backward, $\beta_t(i)$, variables, whose normalized product leads to $\gamma_t(i) = P(q_t = S_i|O_1...O_t)$, *i.e.* the conditional *likelihood* of a particular state S_i at time t, given the observations up to time t, $\{O_1...O_t\}$. The likelihood of a sequence of observations is given by $\mathcal{L} = P(O_1...O_T) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$, where N is the number of hidden states of the HMM. In particular, the expression for the $\alpha_t(i)$ variable is: $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)P_{j|i}]p_j(o_t)$, and for the $\beta_t(i)$ variable: $\beta_t(i) = [\sum_{j=1}^{N} \beta_{t+1}(j)P_{i|j}p_j(o_{t+1})]$, where N is the number of hidden states, $P_{i|j}$ is the transition probability of going to state i given state j and $p_i(o_t)$ is the probability for state i of the observation at time t. From the α and β variables one can obtain the model parameters, *i.e.* the observation and transition probabilities.

The layered formulation of LHMMs makes it feasible to decouple different levels of analysis for training and inference. As we have just shown, each level of the hierarchy is trained independently, with different feature vectors and time granularities. In consequence, the lowest, signal-analysis layer, that is most sensitive to variations in the environment, can be retrained, while leaving the higher-level layers unchanged.

4.2 Inference in LHMMs

The final goal of the system is to decompose in real-time the temporal sequence obtained from the sensors into concepts at different levels of abstraction or temporal granularity. At each level, we use the forward-backward algorithm to compute the likelihood of a sequence given a particular model.

We have implemented two approaches to performing inference with LHMMs. In the first approach, which we refer to as *maxbelief*, the model with the highest likelihood is selected, and this information is made available as an input to the HMMs at the next level. In the *distributional* approach, we pass the full probability distribution over the models to the higher-level HMMs.

As an example, let us suppose that we train K HMMs at level L of the hierarchy, M_k^L, with $k = 1, ..., K$. Let $\mathcal{L}(k)_t^L = \log(P(O(1:t)|M_k^L)) = \log \sum_i \alpha_t(i; M_k^L)$ be the log-likelihood of the observed sequence $O(1:t)$ for model M_k^L; and let $\alpha_t(i; M_k^L)$ be the alpha variable of the standard Baum-Welch algorithm [24] at time t, state i and for model M_k^L, *i.e.* $\alpha_{t+1}(j; M_k^L) = \sum_{i=1}^{N} \alpha_t(i; M_k^L) P_{j|i}^{M_k^L} p_j(o_t; M_k^L)$, where $P_{j|i}^{M_k^L}$ is the transition probability from state j to state i for model M_k^L, and $p_j(o_t; M_k^L)$ is the probability for state j in model M_k^L of observing o_t. At that level, we classify the observations by declaring $C(t)^L = \arg\max_k \mathcal{L}(k)_t^L$, with $k = 1, ..., K$. The next level of the hierarchy $(L + 1)$ could have two kinds of observations of τ temporal length: (1) in the *maxbelief* approach, the hard classification results, $C(1:\tau)^L$, from the previous level for each time step –and therefore a vector of τ discrete symbols in $\{1, ..., K\}$; or (2) in the *distributional* approach, the log-likelihoods, $\{\mathcal{L}(1:K)^L_{t=1}, ..., \mathcal{L}(1:K)^L_{t=\tau}\}$, for each of the models and time instants, –and therefore a vector of K reals for each time step. In our experience, we did not observe performance increases using the latter approach. The results reported in Section 6 correspond to the *maxbelief* approach, which is simpler.

4.3 Decomposition per Temporal Granularity

Figure 1(b) highlights how we decompose the problem into layers with increasing time granularity. For example, at layer L we have a sliding time window of T^L samples. The HMMs at this level analyze the data contained in such time window, compute their likelihood and they generate one observation for layer $L+1$ every T^L samples. That observation is the inferential output of the HMMs in level L, as previously described. The sliding factor along with the window length vary with the granularity of each level. At the lowest level of the hierarchy, the samples of the time window are the features extracted from the raw sensor data (see Section 5.1). At any other level of the hierarchy, the samples are the inferential outputs of the previous level. The higher the level, the larger the time scale—and therefore the higher the level of abstraction—because gathering observations at a higher level requires the outputs of lower layers. In a sense, each layer performs time compression before passing data upward.

Automatic estimation of T^L from data is a challenging problem both for standard HMMs and LHMMs. In the experiments described in this paper, we determined the time granularities at each level based on our intuitions and knowledge about the different classes being modeled at each level. We used cross-validation to select the optimal values from the original set of proposed ones.

5 Implementation of SEER

Focusing on our target application of office awareness, we developed a system named SEER, which employs a two-layer HMM architecture. In this Section we describe in detail the implementation of SEER.

5.1 Feature Extraction and Selection in SEER

The raw sensor signals[3] are preprocessed to obtain feature vectors (*i.e.* observations) for the first layer HMMs.

(1) On the audio side, Linear Predictive Coding (LPC) coefficients [25] are computed. Feature selection is applied on these coefficients by means of principal component analysis (PCA). The number of features is selected such that at least 95% of the variability in the data is maintained, which is typically achieved with no more than 7 features. Other higher-level features are also extracted from the audio signal, such as the energy, the mean and variance of the fundamental frequency over a time window and the zero crossing rate (ZCR) [24], given by: $Zero_s(m) = \frac{1}{N}\sum_{n=m-N+1}^{m}\frac{|sign(s(n))-sign(s(n-1))|}{2}\cdot w(m-n)$, where m is the frame number, N is the frame length, w is a window function, $s(n)$ is the digitized speech signal at an index indicator n, and $sign(s(n)) = \{+1,\ s(n) \geq 0; -1,\ s(n) < 0\}$.

(2) The source of the sound is localized using the Time Delay of Arrival (TDOA) [26] method. In TDOA, one measures the time delays between the signals coming from each sensor. Typically, TDOA-based approaches have two steps: the time delay estimation and the sound source localization. Let $s(n)$ be the source signal and be $x_i(n)$ the signal received by the i-th sensor. If we assume no reverberation, the received signal is given by: $x_i(n) = a_i s(n - t_i) + b_i(n)$. To model reverberation, we add the non-linear reverberation function: $x_i(n) = g_i * s(n - t_i) + b_i(n)$, where a_i is the attenuation factor, b_i is additive noise and g_i is the response between the source and the sensor. In SEER we implemented multiple approaches for estimating the time delay of arrival between the left and right audio signals. We obtained the best performance by estimating the peak of the time cross-correlation function between the left and right audio signals over a finite time window $[N_1, N_2]$, *i.e.*: $r_{lr}(d) = \sum_{n=N_1}^{N_2} l(n)r(n-d)$. This is the method used in the experiments described in Section 6.

(3) On the video side, we divide the video images into four vertical strips of equal width. We extract four features on each strip of the images: the density[4] of skin color in the image strip (obtained by discriminating between skin and non-skin models consisting of histograms in YUV space), the density of motion in the image strip (obtained by image differences), the density of foreground pixels in the image strip (obtained by background subtraction, after having learned the background), and the density of face pixels in the image strip (obtained by means of a real-time face detector to the image [27]).

[3] See Section 3 for a description of the sensors used in SEER.

[4] By "density" we mean the total number of skin pixels in the strip, divided by the total number of pixels in the strip

(4) Finally, a history of the last 1, 5 and 60 seconds of mouse and keyboard activities is logged.

5.2 Architecture of SEER

SEER's architecture is depicted in Figure 2. We employ a two-layer cascade of HMMs with three processing levels. The lowest level captures video, audio, and keyboard and mouse activity, and computes the feature vectors associated to each of these signals (see Section 5.1). These raw sensor signals are processed with time windows of duration less than 100 milliseconds.

First layer HMMs

The first layer of HMMs includes two banks of distinct HMMs for classifying the audio and video feature vectors, with time granularities of less than 1 second. The structure for each of these HMMs is determined by means of cross-validation on a validation set of real-time data.

On the audio side, we train one HMM for each of the following office sounds: *human speech, music, silence, ambient noise, phone ringing*, and the sounds of *keyboard typing*. We will denote this kind of HMMs *discriminative* HMMs. When classifying the sounds, all of the models are executed in parallel. At each instant, the model with the highest likelihood is selected and the sound is classified correspondingly. The source of the sound is also localized, as previously explained (see Subsection 5.1). The video signals are classified using another bank of discriminative HMMs that implement a person detector. At this level, the system detects whether *nobody, one person (semi-static), one active person, or multiple people* are present in the office.

Second layer HMMs

The inferential results[5] from the first layer (*i.e.* the outputs of the audio and video classifiers), the derivative of the sound localization component, and the history of keyboard and mouse activities constitute a feature vector that is passed to the next and highest layer of analysis. The models at this level are also discriminative HMMs, with one HMM per office activity to classify. This layer handles concepts that have longer temporal extent corresponding to a time granularity of about $5 - 10$ seconds. Such concepts include the user's typical activities in or near an office. Office activities recognized by SEER include: (1) PHONE CONVERSATION; (2) PRESENTATION; (3) FACE-TO-FACE

[5] See Section 4 for a detailed description of how we use these inferential results.

conversation; (4) User present, engaged in some other activity; (5) Distant conversation (outside the field of view); and (6) Nobody present. Some of these activities have been proposed in the past as indicators of a person's availability [22].

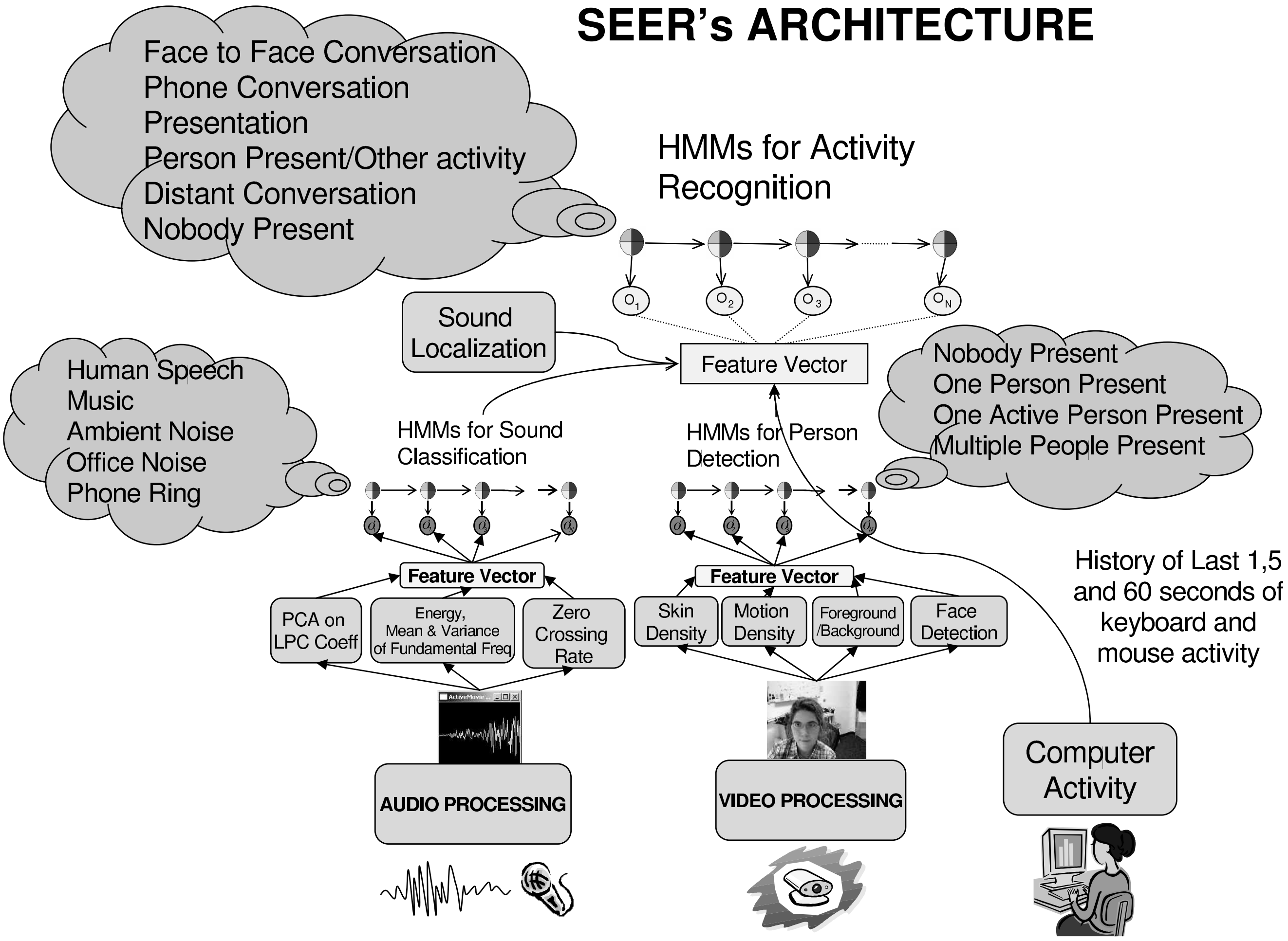


Fig. 2. Architecture of the multimodal Seer system.

6 Experiments

We have tested seer in multiple offices, with different users and respective environments for several weeks. In our tests, we have found that at least the highest layer of seer is robust to changes in the environment. In all the cases, when we moved seer from one office to another, we obtained nearly perfect performance *without* the need for retraining the highest level of the hierarchy. Only some of the lowest-level models required re-training to tune their parameters to the new conditions (such as different ambient noise, background image, and illumination) . The fundamental decomposability of the learning and inference algorithms in LHMMs makes it possible to reuse prior training of the higher-level models, allowing for the selective retraining of layers that are less robust to the variations present in different instances of similar environments.

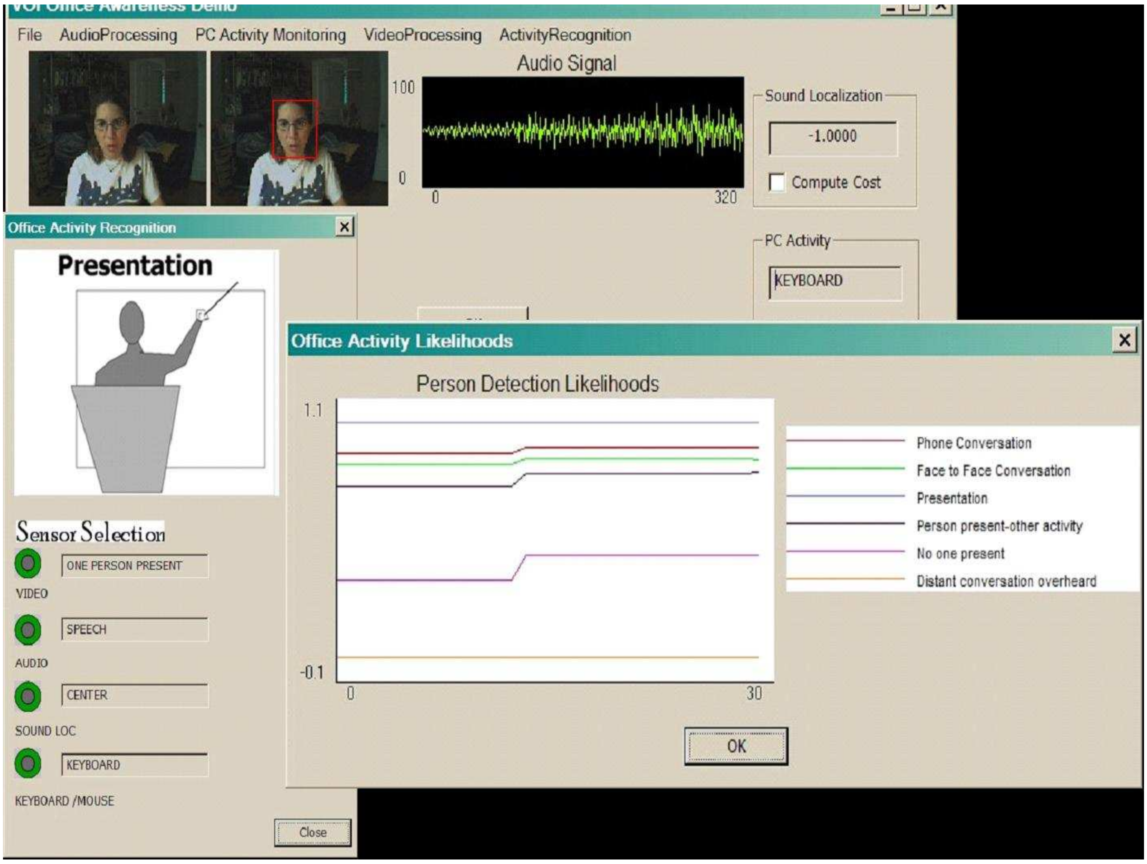
Figure 3 illustrates SEER's user interface and performance in real time. The figure depicts SEER running while a PRESENTATION is taking place in the office. The figure illustrates several features of the user's interface. On the top row, from left to right, we find: (1) The video input. The rectangle around the face on the right image is the output of the face detection module; (2) the raw audio signal; (3) the output of the sound localization module and right below it the output of the computer activity monitoring module. In the second row, from left to right, there is SEER's iconic representation of the activity that has been recognized (PRESENTATION in this case). Right below it there is the value of each of the inputs to the bank of HMMs that performs the office activity recognition (highest layer of HMMs in the LHMM architecture): first the result of the video classification HMMs (*one person present* in the figure); second the result of the audio classification HMMs (*speech* in the figure); third the location of the sound (*center* in the figure); and fourth the value of the keyboard and mouse activity sensor (*keyboard* in the figure). Finally, there is the real-time plot of the normalized likelihoods of the HMMs in the highest layer (i.e. the office activity recognition layer) with their corresponding legend. SEER chooses the model with the highest likelihood as the activity taking place in the office. Note how the highest likelihood model is the one corresponding to PRESENTATION.

In a more quantitative study, we compared the performance of our model with that of single, standard HMMs. The feature vector in the latter case results from the concatenation of the audio, video and keyboard and mouse activity features in one long feature vector. We refer to these HMMs as Cartesian Product (CP) HMMs. For example, in SEER we want to classify 6 different high-level office activities. Let us assume that we use eight-state CP HMMs with single Gaussian observations of dimensionality 16 to model such behaviors. We would need to estimate $8*(16+16+120) = 1216$ parameters for each behavior. An equivalent LHMM with 2 levels would typically have, at the lowest level, two banks of, say, five-state HMMs (six audio HMMs –assuming we have six audio classes, and four video HMMs –assuming four video classes, with dimensionalities 10 and 3 respectively), and at the highest level (the behavior level), a set of 6 four-state HMMs[6] of dimensionality 12, if we use the *distributional* approach: six dimensions for the six audio HMMs, four for the four video HMMs, one for the sound localization component and another for the keyboard and mouse activity history. This amounts to $4*(12+12+66) = 360$

[6] In our experiments the best models obtained using cross-validation had no more than 4 states in LHMMs but needed at least 8 states in the Cartesian Product HMMs.

for each behavior at the second layer. Therefore the number of parameters needed to estimate the office activities is much lower for LHMMs than for CP HMMs. Moreover, in LHMMs the inputs at each level have already been filtered by the previous level and are more stable than the feature vectors directly extracted from the raw sensor data. In summary, encoding prior knowledge about the problem in the structure of the models decomposes the problem in a set simpler subproblems and reduces the dimensionality of the overall model. Therefore, for the same amount of training data, we would expect LHMMs to have superior performance than HMMs. Our experimental results corroborate this expectation.

We point out that it is not considerably more difficult to determine the structure of LHMMs versus that of HMMs. Both for HMMs and LHMMs, we estimated the structure of each of the models—and at each of the levels in LHMMs—using cross-validation. The only additional complexity when designing an LHMM architecture is choosing the number of levels and their respective time granularities. Although this step may be automated in future work, we relied on intuition and knowledge about the domain to handcraft the number of layers and the time granularity of each layer.


File
AudioProcessing
PC Activity Monitoring
VideoProcessing
ActivityRecognition
Audio Signal
100
0
0
320
Office Activity Recognition
Office Activity Likelihoods
Person Detection Likelihoods
1.1
-0.1
0
30
Phone Conversation
Face to Face Conversation
Presentation
Person present-other activity
No one present
Distant conversation overheard
OK
Close

Sound Localization: -1.0000

□ Compute Cost

PC Activity: KEYBOARD

Presentation

Sensor Selection

ONE PERSON PRESENT
VIDEO

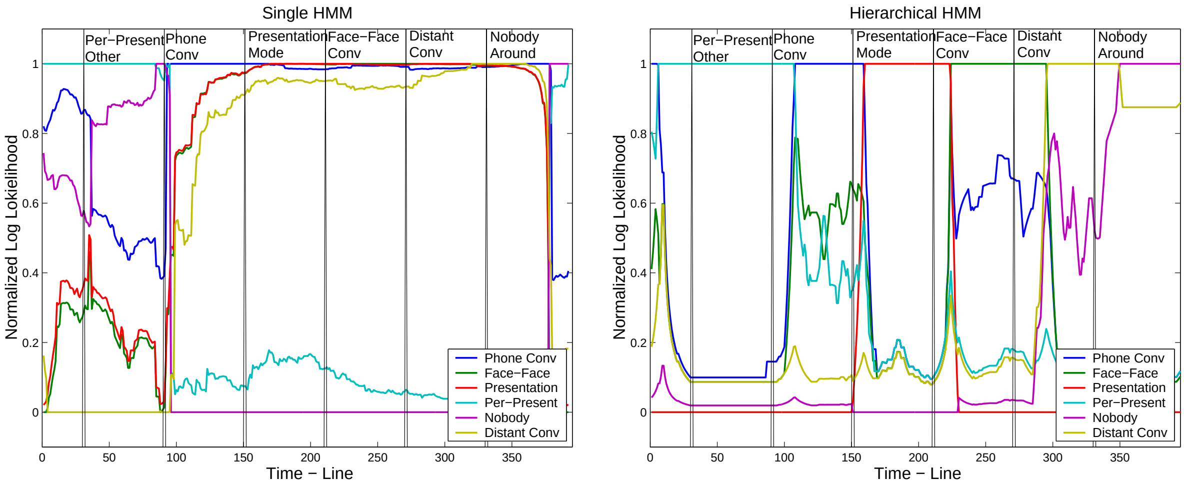
SPEECH
AUDIO

CENTER
SOUND LOC

KEYBOARD
KEYBOARD /MOUSE

Fig. 3. Seer's User Interface while running in real-time during a Presentation

Figure 4 illustrates the per-frame normalized likelihoods on testing in *real-time* both HMMs and LHMMs with the different office activities. By "normalized" likelihoods, we denote the likelihoods whose values have been bounded between 0 and 1. They are given by: $Norm\mathcal{L}_i = \frac{\mathcal{L}_i - min_j(\mathcal{L}_j)}{max_j(\mathcal{L}_j) - min_j(\mathcal{L}_j)}$, for $i = 1, ..., N$, $j = 1, ...N$, and N models. We only plot the likelihoods for the last half of the testing data to avoid instabilities in the transitions. Note that, in the case of LHMMs, the likelihoods are those corresponding to the highest level in the hierarchy, because this is the level that models the office activities.



(a) Single HMMs

(b) LHMMs

Fig. 4. Normalized likelihoods for each of the activity models over time when tested in real time.

Finally, we carried out a different set of experiments. We trained and tested the performance of LHMMs and HMMs on 60 minutes of recorded office activity data (10 minutes per activity, 6 activities and 3 users). Given that it was recorded activity data, we knew the ground truth for each activity. The first few seconds of each dataset were ignored for classification purposes, due to the lag of the models in recognizing each activity. We used 50% of the data –i.e 5 minutes per activity– for training. The rest of the data –*i.e.* 5 min per activity– was used for testing. The results are summarized in Table 1. The average accuracies of both HMMs and LHMMs on testing data were of 72.68% (STD 8.15) and 99.7% (STD 0.95) respectively. In our experience with the system, HMMs normally needed training under similar office conditions (lighting, acoustics, etc.) than that of the particular testing data to obtain reasonable classification results. On the other hand, we can typically reuse the highest level in LHMMs (if not lower layers) that have been trained under different office conditions than that of testing.

Table 1
Confusion matrix for tuned CP HMMs and generic LHMMs on 30 min of real data, where PC=Phone Conversation; FFC=Face to Face Conversation; P=Presentation; O=Other Activity; NA=Nobody Around; DC=Distant Conversation.

Confusion Matrix for highly-tuned HMMs

	PC	FFC	P	O	NA	DC
PC	0.8145	0.0679	0.0676	0.0	0.0	0.05
FFC	0.0014	0.9986	0.0	0.0	0.0	0.0
P	0.0	0.0052	0.9948	0.0	0.0	0.0
O	0.0345	0.0041	0.003	0.9610	0.0	0.0
NA	0.0341	0.0038	0.0010	0.2524	0.7086	0.0
DC	0.0076	0.0059	0.0065	0.0	0.0	0.98

Confusion Matrix for generic LHMMs

	PC	FFC	P	O	NA	DC
PC	1.0	0.0	0.0	0.0	0.0	0.0
FFC	0.0	1.0	0.0	0.0	0.0	0.0
P	0.0	0.0	1.0	0.0	0.0	0.0
O	0.0	0.0	0.0	1.0	0.0	0.0
NA	0.0	0.0	0.0	0.0	1.0	0.0
DC	0.0	0.0	0.0	0.0	0.0034	0.9966

6.1 Discussion

From our experiments we conclude that:

(1) *For the same amount of training data, the accuracy of LHMMs is significantly higher than that of HMMs.* The number of parameters of CP HMMs is higher than that of LHMMs for the office activities being modeled in our experiments. As a consequence, for the same amount of training data, HMMs are more prone to overfitting and worse generalization than LHMMs.

(2) *LHMMs are more robust to changes in the environment than HMMs.* In our experiments, we could not obtain any reasonable performance on CP HMMs had they not been *highly tuned* to the particular conditions of the testing environment. On the contrary, at least the highest level of LHMMs did *not* require retraining, despite the changes in office conditions. This is due to the fact that the CP HMMs carry out high-level inferences about

the user's activity, directly from the raw sensor signals, whereas LHMMs isolate the sensor signals in different sub-HMM models for each input modality.

(3) *The discriminative power of LHMMs is notably higher than that of HMMs.* By *discriminative power*, we mean the distance between the normalized likelihood of the two most likely models. The normalized likelihoods for the HMMs tend to be much closer to each other, making them prone to instability and errors in the classification. Note in Figure 4 how the normalized likelihoods between the two best models in HMMs are much closer than that in LHMMs. This phenomenon is particularly noticeable in the Presentation, Face to Face Conversation, Distant Conversation and Nobody Around activities.

7 Summary, Conclusions and Future Directions

We have described principles and implementation of a real-time, multimodal approach to human activity recognition in an office environment. We have introduced a layered HMM representation (LHMM) that has the ability to capture different levels of abstraction and corresponding time granularities. The representation and associated inference procedure appear to be well matched to the decomposition of signals and hypotheses for discriminating a set of activities in an office setting. Our models are learned from data and can be trained on-the-fly by the user. Some important characteristics of LHMMs when compared to HMMs are: (1) LHMMs encode the hierarchical temporal structure of the discrimination problem; thus, the dimensionality of the state space that needs to be learned from data is much smaller than that of their corresponding CP HMMs; (2) LHMMs, due to their layered structure, are easier to interpret, and, thus, easier to refine and improve, than the corresponding CP HMMs; (3) LHMMs can encode different levels of abstraction and time granularities that can be linked to different levels of representation for human behaviors; (4) the modularity of LHMMs allows the selective retraining of the levels that are most sensitive to environmental or sensor variation, minimizing the burden of training during transfer among different environments.

We have demonstrated the performance of LHMMs in seer, a real-time system for recognizing typical office activities. seer can accurately recognize when a user is engaged in a phone conversation, giving a presentation, involved in a face-to-face conversation, doing some other work in the office, —or when a distant conversation is occurring in the corridor. We believe that our framework can be harnessed to enhance multimodal solutions on the path to more natural human-computer interaction.

We are currently exploring several theoretical and engineering challenges with the refinement of LHMMs. Ongoing work includes our efforts to understand the influence of the layered decomposition on the size of the parameter space, and the resulting effects on learning requirements and accuracy of inference for different amounts of training. Alternate decompositions lead to layers of different configurations and structure; we are interested in understanding better how to optimize the decompositions. We are also exploring the use of unsupervised and semi-supervised methods for training one or more layers of the LHMMs without explicit training effort. Another research direction entails defining different selective perception policies for guiding perception in our models, emphasizing the balance between computation and recognition accuracy. Finally, we are exploring several applications of inference about context.

References

[1] E. Horvitz, Principles of mixed-initiative user interfaces, in: *Proc. CHI'99*, 1999, 159–166.

[2] E. Horvitz, A. Jacobs, D. Hovel, Attention-sensitive alerting, in: *Proc. Conf. on Uncertainty in Artificial Intelligence, UAI'99*, 1999, 305–313.

[3] J. Davis, A. Bobick, The representation and recognition of action using temporal templates, in: *Proc. Computer Vision and Pattern Recognition, CVPR'97*, 1997, 928–934.
URL `citeseer.nj.nec.com/davis97representation.html`

[4] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, in: *Proc. IEEE*, 77 (2), 1989, 257–286.

[5] T. Starner, A. Pentland, Real-time american sign language recognition from video using hidden markov models, in: *Proc. SCV'95*, 1995, 265–270.

[6] A. Wilson, A. Bobick, Recognition and interpretation of parametric gesture, in: *Proc. of International Conference on Computer Vision*, ICCV'98, 1998, 329–336.

[7] M. Brand, V. Kettnaker, Discovery and segmentation of activities in video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[8] A. Galata, N. Johnson, D. Hogg, Learning variable length markov models of behaviour, *International Journal on Computer Vision (IJCV)*, 2001, 398–413.

[9] M. Brand, N. Oliver, A. Pentland, Coupled hidden markov models for complex action recognition, in: *Proc. CVPR'97*, 1996, 994–999.

[10] N. Oliver, Towards perceptual intelligence: Statistical modeling of human individual and interactive behaviors, *Ph.D. thesis, Massachusetts Institute of Technology*, MIT, 2000.

[11] Y. Ivanov, A. Bobick, Recognition of visual activities and interactions by stochastic parsing, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, TPAMI 22(8), 2000, 852–872.

[12] B. Clarkson, A. Pentland, Unsupervised clustering of ambulatory audio and video, in: *Proc. International Conference on Acoustics, Speech and Signal Processing*, ICASSP'99, Vol. VI, 1999, 3037–3040.

[13] F. B. S. Hongeng, R. Nevatia, Representation and optimal recognition of human activities, in: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'00, 2000.

[14] J. Binder, D. Koller, S. J. Russell, K. Kanazawa, Adaptive probabilistic networks with hidden variables, *Machine Learning*, 29 (2-3), 1997, 213–244.

[15] H. Buxton, S. Gong, Advanced Visual Surveillance using Bayesian Networks, in: *Proc. International Conference on Computer Vision*, ICCV'95, Cambridge, Massachusetts, 1995, 111–123.

[16] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, K. Rommelse, The Lumière project: Bayesian user modeling for inferring the goals and needs of software users, in: *Proc. of Fourteenth Conf. in Artificial Intelligence*, 1998, 256–265.

[17] S. S. Intille, A. F. Bobick, A framework for recognizing multi-agent action from visual evidence, in: *Proc. AAAI/IAAI'99*, 1999, 518–525.

[18] A. Madabhushi, J. Aggarwal, A bayesian approach to human activity recognition, in: *Proc. 2nd International Workshop on Visual Surveillance*, 1999, 25–30.

[19] C. Castel, L. Chaudron, C. Tessier, What is going on? a high level interpretation of sequences of images, in: *Proc. workshop on conceptual descriptions from images*, ECCV'96, 1996, 13–27.

[20] J. Fernyhough, A. Cohn, D. Hogg, Building qualitative event models automatically from visual input, in: *Proc. ICCV'98*, 1998, 350–355.

[21] J. Hoey, Hierarchical unsupervised learning of facial expression categories, in: *Proc. ICCV Workshop on Detection and Recognition of Events in Video*, Vancouver, Canada, 2001.

[22] B. Johnson, S. Greenberg, Judging people's availability for interaction from video snapshots, in: *Proc. IEEE Hawaii International Conference on System Sciences*, HICS'99, 1999.

[23] J. Zacks, B. Tversky, Event structure in perception and cognition, *Psychological Bulletin*, 127(1), 2001, 3–21.

[24] L. Rabiner, B. Huang, *Fundamentals of Speech Recognition*, 1993.

[25] J. Deller, J. Proakis, J. Hansen, *Discrete Time Processing of Speech Signals*, Macmillan Series for Prentice-Hall Publishers, New York, 1993.

[26] M. Brandstein, H. Silverman, A practical methodology for speech source localization with microphone arrays, *Computer, Speech and Language*, 11(2), 1997, 91–126.

[27] S. Li, X. Zou, Y. Hu, Z. Zhang, S. Yan, X. Peng, L. Huang, H. Zhang, Real-time multi-view face detection, tracking, pose estimation, alignment, and recognition, *CVPR'01*, Demo summary, 2001.

[28] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden markov model: Analysis and applications, *Machine Learning*, 32 (1), 1998, 41–62.

[29] K. Murphy, M. Paskin, Linear time inference in hierarchical HMMs, in: *Proc. NIPS'01*, 2001.

[30] A. V. Nefian, M. H. HayesIII, An embedded hmm based approach for face detection and recognition, in: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP'99, Vol. vol VI, 1999, 3553–3556.

[31] D. H. Wolpert, Stacked generalization, *Tech. Rep. LA-UR-90-3460*, Los Alamos, NM, 1990.