# Layered Representations for Human Activity Recognition

Nuria Oliver    Eric Horvitz
Adaptive Systems & Interaction
Microsoft Research
Redmond, WA 98052, USA
{nuria,horvitz}@microsoft.com

Ashutosh Garg
Beckman Institute
Univ. Illinois Urbana-Champaign
Champaign, IL 61801
ashutosh@uiuc.edu

## Abstract

*We present the use of layered probabilistic representations using Hidden Markov Models for performing sensing, learning, and inference at multiple levels of temporal granularity. We describe the use of the representation in a system that diagnoses states of a user's activity based on real-time streams of evidence from video, acoustic, and computer interactions. We review the representation, present an implementation, and report on experiments with the layered representation in an office-awareness application.*

## 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 overall *context* of a situation. Although there has certainly been progress on multiple fronts, many challenges remain for developing machinery that can provide rich, human-centric notions of context. Endowing computers with richer notions of context can enhance 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 in Section 2. In Section 3, we describe the challenge of understanding human activity in an office setting, review 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. Context-Sensitive Systems

Location and identity have been the most common properties considered as comprising the user's situation in "context-aware" HCI systems. Context can include other 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 [10].

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. A significant portion of work in this arena has harnessed Hidden Markov Models (HMMs) [16] and extensions. Starner and Pentland in [18] use an HMM for recognizing hand movements used to relay symbols in American Sign Language. More complex models, such as Parameterized-HMM (PHMM) [19], Entropic-HMM [1], Variable-length HMM (VHMM) [8] and Coupled-HMM (CHMM) [2], have been used to recognize more complex activities such as the interaction between two people. Bobick and Ivanov [12], 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 [5]. Brand and Kettnaker in [1] propose an entropic-HMM approach to organize the observed video activities (office activity and outdoor traffic) into meaningful states. In [17], a probabilistic finite-state automaton 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 tend to suffer from a lack of structure, an excess of parameters, and an associated overfitting of data when they are applied to reason about long and complex temporal sequences with limited training data. Finally, in recent years, more complex Bayesian networks have also been adopted for the modeling and recognition of human activities [15, 9, 6, 4, 11, 7].

To date, however, there has been little research on real-time, multimodal systems for HCI that use probabilistic methods to model typical human activities in a hierarchical manner. 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.

## 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. USB camera:** A video signal is obtained via a standard USB camera (Intel), 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 $5$ 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 [13]. We explored the use of Hidden Markov Models (HMMs). Hidden Markov models (HMMs) are 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 [20]. 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. Layered Hidden Markov Models (LHMMs)

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, $O^L = \{O_1^L, O_2^L, ..., O_{T_L}^L\} = O^L(1 : T_L)$, at level $L$, the HMMs at this level, can be thought of as a multiclass classifier mapping these $T_L$ observations to one of $K_L$ classes. Let $\mathcal{X}^{T_L}$ be the 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 $\mathcal{Y}^L = \{1, ..., K_L\}$ 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 $T_{L+1}$, $f_{L+1} : \mathcal{X}^{T_{L+1}} \to \mathcal{Y}^{L+1}$. In this framework, each HMM is learned independently of the others. The availability of labeled data during the training phase allows us to do efficient supervised learning. By itself, each HMM is trained using the Baum-Welch algorithm [16].

The layered formulation of LHMMs makes it feasible to decouple different levels of analysis for training and inference. As we review in Section 5, 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.

We have implemented two approaches to performing inference with LHMMs. In the first approach, which we refer

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

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

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 model $M_k^L$ given all the observations up to time $t$; and let $\alpha_t(i; M_k^L)$ be the alpha variable of the standard Baum-Welch algorithm [16] at time $t$ 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 $\tau$ temporal length: (1) in the *maxbelief* approach, $C(1:\tau)^L$, *i.e.* the hard classification results from the previous level for each time step –and therefore a vector of $\tau$ discrete symbols in $\{1, ..., K\}$ ; or (2) in the *distributional* approach, $\{\mathcal{L}(1:K)_{t=1}^L, ..., \mathcal{L}(1:K)_{t=\tau}^L\}$, *i.e.* the log-likelihoods for each of the models and time instants, –and therefore a vector of $K$ reals for each time step. In our experience, we didn't observe performance increases using the latter approach. The results reported in section 6 correspond to the *maxbelief* approach, which is simpler.

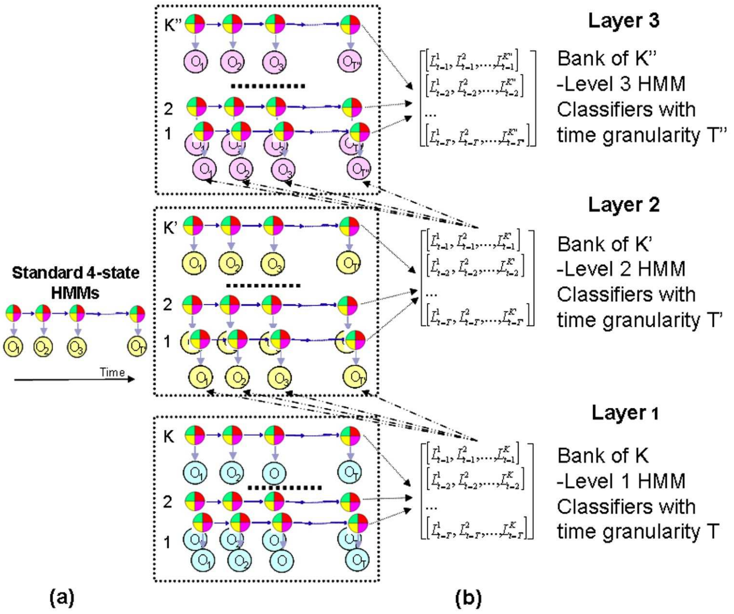


**Figure 1. Graphical representation of (a) HMMs, and (b) LHMMs with 3 different levels of temporal granularity.**

### 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 every 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). 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.

Focusing on our target application of office awareness, we employ a two layer HMM architecture. The raw sensor signals are processed with time windows of duration less than $100$ milliseconds. Next, the lowest layer of HMMs classify the audio and video data with a time granularity of less than $1$ second. The second layer of HMMs represents typical office activities, associated with a time granularity of about $5-10$ seconds. The activities modeled in this setting are: (1) PHONE CONVERSATION; (2) PRESENTATION; (3) FACE-TO-FACE CONVERSATION; (4) USER PRESENT, ENGAGED IN SOME OTHER ACTIVITY; (5) DISTANT CONVERSATION (outside the field of view); (6) NOBODY PRESENT.

## 5. Implementation of SEER

We explored the use of LHMMs in a system named SEER, which employs a two-layer HMM architecture.

### Feature Extraction and Selection in SEER

The raw sensor signals are preprocessed to obtain feature vectors (*i.e.* observations) for the first layer of HMMs. With respect to the audio analysis, Linear Predictive Coding coefficients [16] are computed. Feature selection is applied to these coefficients via principal component analysis. 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. We also extract other higher-level features from the audio signal such as its energy, the mean and variance of the fundamental frequency over a time window, and the zero crossing rate [16], given by $Z_s(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|sign(s(n)) - sign(s(n-1))|}{2} \cdot w(m-n)$, where $m$ is the frame number, $N$ is the frame length, $w$ is a window function, $s(n)$ is the digitized audio signal at an index indicator $n$, and $sign(s(n)) = \{+1,\ s(n) \geq 0; -1,\ s(n) < 0\}$.

The source of the sound is localized using the Time Delay of Arrival (TDOA) method. In TDOA [3], 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. 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.

With respect to the video, four features are extracted from the video signal: the density of skin color in the image (obtained by discriminating between skin and non-skin models, consisting of histograms in HSV color space), the density of motion in the image (obtained by image differences), the density of foreground pixels in the image (obtained by background subtraction, after having learned the background), and the density of face pixels in the image (obtained by means of a real-time face detector [14]).

Finally, a history of the last $5$ seconds of mouse and keyboard activities is logged.

**Architecture of SEER**

We employ a two-level cascade of HMMs with three processing layers. The lowest layer captures video, audio, and keyboard and mouse activity, and computes the feature vectors associated to each of these signals (see Section 5).

The middle layer includes two banks of distinct HMMs for classifying the audio and video feature vectors. 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 explained before. The video signals are classified using another set of 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.

The inferential results[3] from this 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 (third) and highest layer of analysis. This layer handles concepts with longer temporal extent. Such concepts include the user's typical activities in or near an office. The models at this level are also discriminative HMMs.

**Learning in SEER**

For an HMM, the problem of learning the model parameters is solved by the *forward-backward* or Baum-Welch algorithm [16]. This algorithm provides expressions for the forward, $\alpha_t(i)$, and backward, $\beta_t(i)$, variables, whose normalized product leads to $\gamma_t(i) = P(q_t = S_i | O(1:t))$, *i.e.* the conditional $likelihood$ of a particular state $S_i$ at time $t$, given $O(1:t)$, *i.e.* the observations up to time $t$. The log-likelihood of a sequence of observations is given by $\mathcal{L} = \log P(O(1:T)) = \log \sum_{i=1}^{N} \alpha_T(i)$, where $N$ is the number of hidden states of the HMM. In particular, the expressions for the $\alpha_t(i)$ and $\beta_t(i)$ variables are $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) P_{j|i}] p_j(o_t)$, and $\beta_t(i) = [\sum_{j=1}^{N} \beta_{t+1}(j) P_{i|j} p_j(o_{t+1})]$, where $N$ is the number of hidden states, $P_{i|j}$ is the probability of state $i$ given state $j$ and $p_i(o_t)$ is the probability for state $i$ of observing $o_t$. From the $\alpha$ and $\beta$ variables one can obtain the model parameters (the observation and transition probabilities).

**Classification in SEER**

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. As the classifiers at each level are a set of HMMs, we adopt standard HMM inferencing techniques. We use the forward-backward algorithm to compute the likelihood of a sequence given a particular model at a particular level.

## 6. Experiments

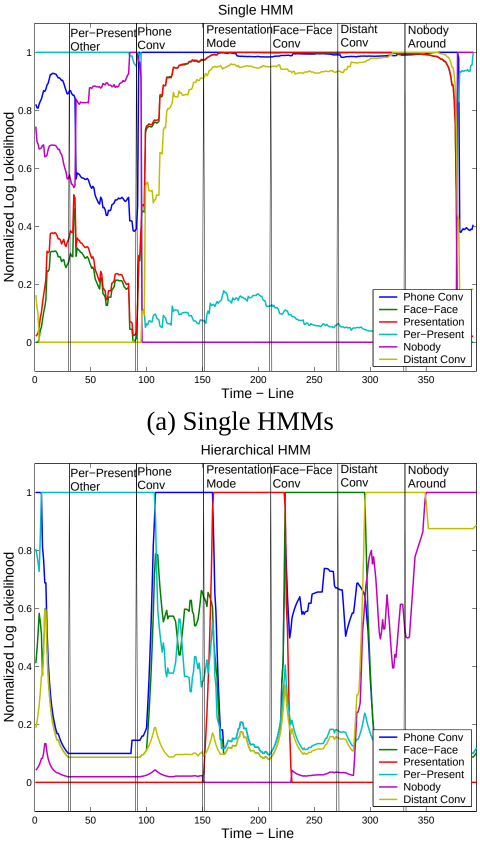
We have tested SEER in multiple offices, with different users and respective environments for several weeks. In our tests, we have found that the high-level layers of SEER are relatively robust to changes in the environment. In all the cases, when we moved SEER from one office to another, we obtained nearly perfect performance *without* the need for retraining the higher levels 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 of 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.

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/mouse activities features in one long feature vector. We refer to these HMMs as the Cartesian Product (CP) HMMs. The number of parameters to estimate 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

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

HMMs and LHMMs, we estimated the structure of each of the models—and at each of the levels for LHMMs—using cross-validation. The only additional complexity when designing an LHMM architecture is choosing the number of levels and their time respective 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.

Figure 2 illustrates the per-frame normalized [4] likelihoods on testing in *real-time* both HMMs and LHMMs with the different office activities. 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

**Figure 2. Log-likelihoods for 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. In particular, we used about $20$ sequences of each class for the audio and video HMMs (first layer) and $10$ sequences of each office activity for the behavior HMMs (second layer). 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 need 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 (**PC=Phone Conversation; FFC=Face to Face Conversation; P=Presentation; O=Other Activity; NA=Nobody Around; DC=Distant Conversation).

**Confusion Matrix for tuned CP 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 the 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 limited 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, the HMMs were more sensitive to changes in the environment than LHMMs. We could not obtain reasonable performance on the CP HMMs had they not been *tuned* to the particular testing environment and conditions. test them under some particular conditions. On the contrary, at least the highest layer of our 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.

[4] By "normalized" likelihoods, we denote the likelihoods whose values have been bounded between 0 and 1. They are given by: $NormL_i = \frac{\mathcal{L}_i - min_j(\mathcal{L}_j)}{max_j(\mathcal{L}_j) - min_j(\mathcal{L}_j)}$, for $i, j = 1, \ldots, N$, and $N$ models.

**3.** *The discriminative power of LHMMs is notably higher than that of HMMs*. By *discriminative power*, we mean the distance between the log-likelihoods of the two most likely models. The log-likelihoods for the CP HMMs tend to be much closer to each other, making them prone to instability and errors in the classification. Note in Figure 2 how the normalized likelihoods between the two best models in CP HMMs are much closer than that in LHMMs.

## 7. Summary and Future Directions

We have presented a representation, reasoning principles and an implementation of a real-time, multimodal approach to human activity recognition in an office environment. We have focused on properties of a layered HMM (LHMM) methodology 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 can encode the hierarchical temporal structure of the office activity modeling problem; (2) LHMMs, due to their layered structure, are easier to interpret, and, thus, easier to refine and improve, than the corresponding CP HMMs; (3) the dimensionality of the state space that needs to be learned from data is smaller in LHMMs than that of their corresponding CP HMMs; in consequence, LHMMs are less prone to overfitting than HMMs; (4) LHMMs can encode different levels of abstraction and time granularities that can be linked to different levels of representation for human behaviors; (5) 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 carried out experiments probing 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 LHMMs can be used to enhance multimodal solutions on the path to more natural human-computer interaction.

We are currently exploring the refinement of LHMMs along several dimensions. We are pursuing a deeper understanding of 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. Such an understanding could enable us to optimize the decompositions. We are also comparing our LHMMs representation to other hierarchical representations, and exploring the use of unsupervised and semi-supervised methods for training one or more layers of the LHMMs without explicit training effort.

## References

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

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proc. of CVPR97*, pages 994–999, 1996.

[3] M. Brandstein and H. Silverman. A practical methodology for speech source localization with microphone arrays. 11(2):91–126, 1997.

[4] H. Buxton and S. Gong. Advanced Visual Surveillance using Bayesian Networks. In *International Conference on Computer Vision*, pages 111–123, Cambridge, Massachusetts, June 1995.

[5] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *International Conference on Acoustics, Speech and Signal Processing, ICASSP'99*, volume VI, pages 3037–3040, 1999.

[6] J. Fernyhough, A. Cohn, and D. Hogg. Building qualitative event models automatically from visual input. In *ICCV'98*, pages 350–355, 1998.

[7] J. Forbes, T. Huang, K. Kanazawa, and S. Russell. The batmobile: Towards a bayesian automated taxi. In *Proc. Fourteenth International Joint Conference on Artificial Intelligence, IJCAI'95*, 1995.

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

[9] J. Hoey. Hierarchical unsupervised learning of event categories, Unpublished Manuscript, 2001.

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

[11] S. S. Intille and A. F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI/IAAI'99*, pages 518–525, 1999.

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

[13] B. Johnson and S. Greenberg. Judging people's availability for interaction from video snapshots. In *Proc. of the IEEE Hawaii International Conference on System Sciences, HICS'99*, 1999.

[14] S. Li, X. Zou, Y. Hu, Z. Zhang, S. Yan, X. Peng, L. Huang, and H. Zhang. Real-time multi-view face detection, tracking, pose estimation, alignment, and recognition, 2001.

[15] A. Madabhushi and J. Aggarwal. A bayesian approach to human activity recognition. In *In Proc. of the 2nd International Workshop on Visual Surveillance*, pages 25–30, 1999.

[16] L. Rabiner and B. Huang. *Fundamentals of Speech Recognition*. 1993.

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

[18] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceed. of SCV'95*, pages 265–270, 1995.

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

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