Multimodal Estimation of User Interruptibility for Smart Mobile Telephones

Robert Malkin, Datong Chen, Jie Yang, Alex Waibel
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave.
Pittsburgh PA USA 15213
[rgmalkin,datong,yang+,ahw]@cs.cmu.edu

ABSTRACT

Context-aware computer systems are characterized by the ability to consider user state information in their decision logic. One example application of context-aware computing is the smart mobile telephone. Ideally, a smart mobile telephone should be able to consider both social factors (i.e. known relationships between contactor and contactee) and environmental factors (i.e. the contactee's current locale and activity) when deciding how to handle an incoming request for communication.

Toward providing this kind of user state information and improving the ability of the mobile phone to handle calls intelligently, we present work on inferring environmental factors from sensory data and using this information to predict user interruptibility. Specifically, we learn the structure and parameters of a user state model from continuous ambient audio and visual information from periodic still images, and attempt to associate the learned states with user-reported interruptibility levels. We report experimental results using this technique on real data, and show how such an approach can allow for adaptation to specific user preferences.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms

Keywords

Smart mobile telephones, user interruptibility, context awareness, HMMs, hierarchical HMMs, scene learning

1. INTRODUCTION

Context-aware computer systems are chararcterized by the ability to consider user state information in their decision logic. One application of context- aware computing is the smart mobile telephone. Standard mobile telephones provide a constant, instant communications channel, allowing human users to stay connected with one another and achieve tremendous levels of efficiency in both vocational and social settings. However, by virtue of the fact that they are always on unless explicitly switched off, they also present opportunities for annoyance, unwanted interruption, and distraction. Many users find incoming calls disruptive under certain conditions: during meetings or seminars, while driving, while attending theatrical performances, or during meals. Further, under certain adverse conditions, such as in the proximity of a construction site, participating in a conversation may be physically difficult. Toward the goal of alleviating these problems, researchers have begun to apply context-aware computing techniques to the mobile telephone platform; see for example work by Danninger et. al. [3], [4], and Siewiorek et. al. [19].

Ideally, a smart mobile telephone should be able to consider both social factors (i.e. known relationships between contactor and contactee) and environmental factors (i.e. the contactee's current locale and activity) when deciding how to handle an incoming request for communication — by ringing, vibrating, taking a message, giving or withholding information about the contactee's state, or even scheduling a more convenient time for the communication to take place.

In this research, we focus on modeling and detecting environmental and activity factors affecting interruptibility. Using hierarchical models of user state learned in an unsupervised fashion from raw sensory data, we estimate whether or not the contactee is interruptible. Combined with social information and a means of integrating these two information sources to form a call-handling logic, this approach moves toward the goal of a smart mobile telephone.

The remainder of this paper is organized as follows. We discuss the theoretical and practical aspects of modeling user interruptibility in Sec. 2. We show how to integrate sensory and prior information into a multimodal interruptibility model in Sec. 3. Sec. 4 contains a detailed account of the experiments we conducted to test our approach. We discuss methods for adapting this model to both new or changing user preference and novel environmental conditions in Sec. 5. Our conclusions are found in Sec. 6.

2. MODELING USER INTERRUPTIBILITY

Our interruptibility model encompasses the following variables and sets of dependencies. I represents interruptibil-

ICMI '06 Banff, Alberta, Canada

ity; in this study, a binary-valued variable. C_O represents directly observable context features; these include the identity of the contactor and contactee activities that the system knows about from calendar entries, for example, meetings. C_H represents hidden context information; this information must be inferred from the observable acoustic evidence E_A and the observable visual evidence E_V. C_H is itself composed of two variables; C_E represents the contactee's immediate environment (e.g. office or city street), while C_A represents the contactee's current activity (e.g. preparing a report or hailing a cab). The dependencies among these variables are shown in Eqns. 1 - 5.

$$C_E \leftrightarrow C_A, \tag{1}$$

$$E_A \leftarrow C_E, C_A, \tag{2}$$

$$E_V \leftarrow C_E, C_A, \tag{3}$$

$$C_H \leftarrow E_A, E_V, \tag{4}$$

$$I \leftarrow C_H, C_O. \tag{5}$$

In this work, we consider only the hidden user state information C_H. We next consider how to model C_H using observable acoustic and visual evidence, as well as how to model I using information from C_H.

2.1 Inferring User State from Sensory Data

Seiwiorek et. al. used a simple interruptibility model involving only a few sensory features; notably two audio signal power levels (one from a microphone capturing mainly contactee speech, the other capturing mainly ambient noise) and visual light levels [19]. This model, while useful, does not capture some important aspects of user state. First, the audio features focus mainly on conversation; the assumption is that users do not want to be interrupted while they are already involved in face-to-face or telephone conversations. While this assumption appears to hold on average, it may not always be the case. Second, specific patterns of activity and interruptibility, including those which are stable and repeated over time, are not accounted for. These patterns, when identified on a per-user basis, can be used to improve interruptibility assessments.

Hudson, Fogarty, et. al. focused in [9] on the predictive power of high-level sensors, such as "talking on telephone" and "sitting at monitor" in a Wizard-of-Oz study in an office environment and acheived promising results in this domain. They further demonstrated in [**?**] that real sensors were able to perform quite well under real conditions using a combination of audio, visual, and computer interaction features. Horvitz and Apacible also demonstrated in [8] the use of audiovisual sensors for estimting interruptibility in the office domain; their model explicitly attempted to model the cost of an interruption as another information source.

These previous studies focused on a stationary setting. In [3], Danninger et. al. modeled user state in a mobile setting given ambient acoustic information solely in terms of environments. As shown in work by Ellis and Lee ([5], [6]) and Malkin ([10], [11]), a low-resolution approach can be used to model environments. In that user context depends on environment, and that environment and activity are mutually dependent (Eqn. 1), this approach does to some extent capture the information that we are interested in. One might argue, though, that it is really user *activities* that matter in this application. For example, a user simply walking down a city street might be interruptible while a user walking down a city street while engaged in a conversation might not. A low-resolution environment-based context model might correctly spot the city street, but miss the conversation and thus fail to make the interruptibility distinction.

There is an existing body of work on inferring user activity from sensory data employing multiresolution models; see work from Clarkson and Pentland ([1], [2]) and Oliver et. al. ([14]). The models employed in these works are known as layered hidden Markov models (LHMMs). LHHMs consist of multiple HMMs with varying temporal resolutions and modeling capacities arranged in series such that the output of a low-level HMM serves as the input of a high-level HMM. In this way, the LHMM can model high-level activities or scenes as compositions of lower-level events. There are some issues with LHMMs for this type of task, however; we now discuss these problems and explain why a slightly different multiresolution model, the hierarchical hidden Markov model (HHMM) is a better choice for modeling multilayered sensory phenomena.

We use two sensory streams in this work; continuous audio data and sequences of still images. We elect to use these two modalities as opposed to a more detailed modality like video for two reasons. First, we are studying the ability to detect interruptibility in a *mobile* environment. As such, it is important to recognize that we are inherently limited both in sensory capacity and in computational power available. We strongly suspect that most users would reject a system that required any equipment other than a smartphone; it is thus necessary to forego full video and focus on audio plus still images, which are much cheaper to collect, store, and process. Second, it has been demonstrated by many of the researchers noted above that continuous audio is a rich source of information for activity recognition. Adding still images to audio may not enrich the sensory stream to the same degree that full video would, but provided images are captured often enough (i.e. more often than human activities change), the gain from adding full video would be minimal.

2.2 A Hierarchical Sensory Context Model

The LHMMs presented in the works cited above have been successfully employed to model a variety of multiresolution phenomena. They do have some shortcomings, however. First, the Viterbi procedure for inference must be executed once for each model level, as output from lower levels (either in the form of discrete state sequences or some equivalent continuous feature space) is required for high-level input. In addition to this inefficiency, the LHMM obscures the fact that dependencies flow from causes to effects (i.e. from the top down) rather than from effects to causes (i.e. from the bottom up). To address these two concerns, we use the HHMM introduced by Fine et. al. in [7]. Murphy showed in [13] and [12] that inference in the HHMM could be done in linear time (relative to input length) using the junction tree algorithm, improving on the cubic time algorithm proposed by Fine. Xie et. al. additionally showed in [21] and [22] a practical method for compiling an HHMM down to a standard HMM, also allowing for inference in linear time (though with potentially many more states than Murphy's algorithm).

Murphy additionally provided the outlines of an approach for learning HHMM structure from data; Xie et. al. dis-

cussed in detail a randomized, top-down approach for learning two-level structures from soccer videos in which the model is, at each step, either expanded, pruned, shuffled, or optimized with EM and probabalistically accepted or rejected using a Bayesian information criterion (BIC) ratio.

We propose a simple, bottom-up method for learning hierarchical structures of arbitrary depth. This method involves an initialization step, in which the data are clustered into short-timespan event models, and a scene learning step, in which sequences of events are combined to form long-timespan scenes. We discuss each of these steps in turn below.

2.2.1 Initialization

There are several options for clustering sensory data into events. Here, we consider three clustering algorithms and discuss their advantages and disadvantages. In all that follows, we denote by k the number of models to be initialized, by n the number of states assigned to each model, by t the number of frames we assign to a model state for initialization, by f a corpus of sensory features, by S a set of sensory segments of length $n \times t$, by M a set of initialized models, and by g a fully connected grammar constructed from the models in M.

Perhaps the simplest approach to model initialization, the segmental k-means (SKM) algorithm, was popularized by Clarkson and Pentland in [1]. The SKM algorithm, shown in Fig. 1, requires several parameters to be set by hand; most importantly, the number of models k with which to partition the data. Given this constraint, SKM is fast and conceptually simple.

An alternate approach, when the number of models k is not known in advance or cannot be estimated using knowledge of the problem, is to use some type of leader-follower clustering algorithm such as the k-variable k-means (KVKM) algorithm suggested by Reyes-Gomez and Ellis in [16]. This algorithm, introduced for HMM topology selection in the domain of general sound modeling, creates an initial model and subsequently creates new models as warranted by the data. We extend this algorithm to our problem, resulting in the k-segment k-means (KSKM) algorithm, shown in Fig. **??**. The disadvantages of this algorithm are poor speed and sensitivity to data order and model spawning parameters.

Finally, one can use an agglomorative clustering procedure such as the one suggested by Slaney [20] and shown in Fig. 3. This approach suffers the same disadvantages as the KSKM algorithm; additionally, a stopping point must be supplied in an application-dependent manner.

Given that all three algorithms require some application dependent knowledge, we opt in this work for the simplest approach, the SKM algorithm.

2.2.2 Scene Identification

In the LHMM approach, the state sequence inferred from lower-level features is converted into a new feature space for a higher-level HMM. The SKM algorithm is then used to build a model set over this higher-level feature space, which can be either a sequence of discrete state identities or a vector of state posteriors. This process can be repeated until the desired number of levels is reached. The multi-level SKM approach could also be used to identify scenes for an HHMM by replacing the outputs of each high-level state with vertical transitions to low-level states. The main drawback of this

Segmental K-Means

Given: k, n, t, f:
Initialize: choose k samples S to initialize M.
repeat until convergence
    Build grammar g from M.
    Segment f using g.
    EM training until parameters converge.

Figure 1: The Segmental K-Means Algorithm

K-Segment K-Means

Given: n, t, f,
θ_m, a merging threshold,
θ_s, a spawning threshold,
z, the minimum number of samples per model.
Initialize: choose segment S_0, train model M_0.
repeat until likelihood $P(S|M)$ converges:
    for each unassigned sample S_i:
    Find model M_j with highest $P(S_i|M_j)$.
    if $P(S_i|M_j) > \theta_m$:
        then Add S_i to M_j.
    elseif $P(S_i|M_j) < \theta_s$:
        then create new model M_i using S_i.
    Update $P(S|M)$.
Remove models M_i with fewer than z samples.
repeat until convergence
    Build grammar g from M.
    Segment f using g.
    EM training until parameters converge.

Figure 2: The K-Segment K-Means Algorithm

Agglomorative Segment Clustering

Given: n, t, f,
θ, a merging threshold.
Initialize: train model M_i for each segment S_i.
repeat
    for each model pair (M_i, M_j) :
    $\Phi_{i,j} \leftarrow L(S_i|M_j) + L(S_j|M_i)$.
    if $\Phi_{i,j} > \theta$ for best pair **return**
    Build new model M_k with data $S_k = (S_i, S_j)$.
    Remove M_i, M_j from M.

Figure 3: The Agglomorative Segment Clustering Algorithm

method is that every low-level state must be part of some high-level scene. There may in fact be many low-level events which appear essentially at random intervals. By forcing these events into scenes in a maximum likelihood manner, we obscure the fact that they have no real predictive power.

To address this problem, we propose a method similar to one used to learn phrases and letter sequences in natural text; see for example work by Ries et. al. [17] or Ron et. al. [18] on variable length Markov models. In this method, which we call the minimum mutual information (MMI) method, we define a scene as a set of events which predictably occur in the same temporal neighborhood. That is, if the appearance of one specific event a significantly increases the likelihood that some other event b will occur in the near future, then a and b should be grouped together as part of the same scene c. To turn this observation into an algorithm, we note that the relationship between events a and b described above is one of temporal dependency. We can measure the degree to which a corpus C of length T with symbols drawn from alphabet S contains temporal dependencies between adjacent symbols C_t and C_{t-1} by computing its average temporal mutual information $I(C)$, as shown in Eqn. 6. In this work, we find it convenient to normalize this metric to obtain a figure between 0 and 1; this normalized measure, $\hat{I}(C)$, is given in Eqns. 7 and 8.

$$I(C) = I(C_t; C_{t-1}) = H(C_t) - H(C_t|C_{t-1}), \quad (6)$$

$$\hat{I}(C) = \hat{I}(C_t; C_{t-1}) = \frac{I(C_t; C_{t-1})}{H(C_t)}, \quad (7)$$

$$= 1 - \frac{H(C_t|C_{t-1})}{H(C_t)}. \quad (8)$$

When a corpus has high temporal mutual information, it means that the next symbol is easily predictable given the present symbol. Put another way, such a corpus has high temporal redundancy and is thus represented inefficiently. Given that the goal in perceptual systems is usually to reduce redundancy and represent sensory data efficiently, we can use the temporal mutual information measure to learn scenes. Specifically, if we define a "scene" as any sequence of symbols in a corpus which are redundant, then removing such sequences from the corpus and replacing them with some higher-level symbol, $\hat{I}(C)$ is reduced. Given this observation, we define the MMI structure learning algorithm as in Fig. 4; here, ψ refers to any additional constraints placed on the merging procedure; typical constraints include minimum co-occurrence counts, restrictions on the number of children a new symbol can have, restrictions on whether or not all level merges at level n must be exhausted before considering merges at level $n+1$, and the like. These restrictions can have a large effect on the final topology of the learned HHMM. For instance, if a state can have few children, the topology will be fairly deep, as the only way to learn longer structures is to build vertically. Conversely, if a state can have many children, it is possible to learn long structures by building horizontally, resulting in a shallower topology.

After completing the merging procedure in this way, each candidate merge is converted into a hierarchical HMM structure; every new symbol S_q becomes an HHMM state which spawns all states it replaced with the appropriate prior, transition, and exit probabilities. The remaining symbols which were not merged into a scene can be merged into a special "no scene" symbol, or simply left unmerged.

MMI Structure Learning

```
Given: C, S, ψ:
Initialize: Compute Î(C) = Î(C_t; C_{t-1}).
repeat
        M ← {}, C' ← C, Î(C*) ← Î(C).
        for each symbol pair (S_i, S_j) :
        if ψ(S_i, S_j) = TRUE :
          then
                Create new symbol S_q.
                C' ← sS_i + S_j + S_q.
                Compute Î(C') = Î(C'_t; C'_{t-1}).
                if Î(C') < Î(C*) :
                  then
                        M ← (S_i, S_j), C* ← C',
                        Î(C*) ← Î(C').
if Î(C*) < Î(C) :
   then
        C ← C*, Î(C) ← Î(C*).
   else return
```

Figure 4: The Minimum Mutual Information Structure Learning Algorithm

3. AN AUDIO-VISUAL CONTEXT MODEL

Given a method for learning scenes from data, and hence for inferring state sequences from data, we consider how to turn this information into an estimate of user interruptibility. That is, if we infer from data some state sequence S^*, we wish to know the value of the binary interruptibility variable I during that sequence. More accurately, since this interruptibility estimate will ultimately be combined with estimates from non-sensory modalities, we wish to estimate the probability $P(I|S^*)$ for all values of I. Using Bayes' rule and borrowing from automatic speech recognition (ASR) the engineering convention of weighting the prior and the likelihood, we show how to estimate $P(i|S^*)$ for some value $i \in I$ in Eqns. 9 - 13.

$$P(i|S^*) \propto P(S^*|i)P(i), \quad (9)$$

$$= \prod_{t=1}^{T} P(S_t|i)P(i), \quad (10)$$

$$= TP(i)\prod_{t=1}^{T} P(S_t|i), \quad (11)$$

$$\approx \alpha P(i)\frac{\beta}{T}\prod_{t=1}^{T} P(S_t|i), \quad (12)$$

$$\log P(i|S^*) \approx \log \alpha P(i) + \log \beta \sum_{t=1}^{T} P(S_t|i). \quad (13)$$

Both the interruptibility model $P(S^*|I)$ and the interruptibility prior $P(I)$ can by learned by simple frequency counting of inferred states combined with user-supplied interruptibility labels.

Subcorpus	Audio Time	Images	Min. / Image
1	7836s	20	6.5
2	19320s	44	13.0
3	22282s	30	12.3
Total	49438s	94	11.4

Table 1: Evaluation Corpus

In this work, we aim to consider both auditory and visual information when estimating interruptibility. Rather than attempting to merge auditory and visual features into a single feature vector, we opt for late fusion. We can thus compute a separate score for each modality and combine them in a weighted fashion as shown in Eqn. 14. Here we assume that the auditory and visual information are independent. This assumption does not always hold; as shown in our depedency model above (Eqns. 1 - 5), both are related to environment and activity. We make this independence assumption in the interest of simplifying the model.

$$P(i|S_A^*, S_V^*) = \lambda P(i|S_A^*) + (1-\lambda)P(i|S_V^*). \quad (14)$$

4. EXPERIMENTS

We conducted a set of experiments designed to test our method's ability to predict interruptibility from auditory and visual data. Since we have limited visual information, we divide our experiments into two sets: one set designed to find the best acoustic model for interruptibility, and one set designed to find the best combination of acoustic and visual models. For both experiments, we induced user context model structures using all available data. We then conducted round-robin training and testing of state interruptibility models and prior interruptibility models in which two of the three recordings were used for training and the third for testing. After describing our corpus in Sec. 4.1, we discuss audio and visual feature extraction in Sec. 4.2, followed by a dis- cussion of the parameters used to construct the acoustic user context model in Sec. 4.3. Sec. 4.4 discusses the construction of the visual context model, while Secs. 4.5 and 4.6 contain experimental results.

4.1 Experimental Corpus

The data we used in this study was collected by one of the authors as he carried acoustic and visual sensors during normal daily activities. Audio was captured using the Neuros II personal audio computer in conjunction with a Sony ECM-719 stereo microphone and a portable, battery-powered preamplifier from SoundProfessionals. Audio was captured at 2-byte sample depth at 48kHz and later downsampled to 16kHz. One channel was used. Visual information was captured by periodic VGA-quality snapshots from the camera on a Nokia 6600 mobile telephone. Pictures were taken, on average, every 11 minutes, though the rate of photos varied with activity; more shots were taken when the scene was changing rapidly and fewer were taken during those periods where the author was mainly sitting at his desk. We collected nearly 14 hours of data and 100 images; details are shown in Tab. 1. In addition to serving as visual input, the images were also used to label the corpus for interruptibility

4.2 Feature Extraction

We extracted 11 mel-frequency cepstral coefficients (MFCCs) from the audio signal at a rate of 100 frames per second. We extracted three additional features to supplement the MFCCs. These additional features included spectral centroid (a measure of the perceptual "brightness" of the signal), spectral diffusion (which measures the spread of spectral energy in frequency space), and signal-to-noise ratio (which helps to distinguish noisy environments from merely loud ones). After merging these features into a single 14-dimensional acoustic feature vector, we filtered them by applying a Gaussian smoothing window. Finally, we normalized each feature globally to zero mean and unity variance.

Visual information was characterized for these experiments by local features and the correlations among local features. We extracted three types of local features from $4 \times 3 = 12$ regular granularities of each image. In each local image patch, we extracted the mean of grayscale values, the means of R, G, and B values (the redundant information here is to emphasize grayscale values), and the 24-bin color histogram in HSV color space. Since there are 12 patches in each image, the grayscale mean vector has 12 dimensions represented by column vectors V_g. The mean of RGB values is represented as a 12×3 matrix V_{rgb} and the color histogram is denoted 12×24 matrix V_h.

The correlations among the local features characterized how local patches were similar to each other. We computed the self-correlation matrices for each type of local feature by using the definitions of the grayscale mean correlation matrix M_g:

$$M_g = V_g V_g^T, \quad (15)$$

the RGB mean correlation matrix M_{rgb}:

$$M_{rgb} = V_{rgb} V_{rgb}^T, \quad (16)$$

and the color histogram correlation matrix M_h:

$$M_h = V_h V_h^T. \quad (17)$$

The final visual feature vecotr for an image is the combination of the local features and their correlations, which is formally defined as:

$$F_v = [V_g, V_{rgb}, V_h, M_g, M_{rgb}, M_h]. \quad (18)$$

4.3 Building the Acoustic User Context Model

To build the hierarchical acoustic user context model, we first had to infer from the data a suitable event-level model. To do this, we used all the data listed in Tab. 1. We used the SKM algorithm with $k = 32$ models and $n = 3$ states per model. We observed that the average event duration after final segmentation was heavily dependent on the inter-model transition penalty, and less dependent on the number of initial frames assigned to each state. Tab. 2 shows how these two parameters affected average event length.

Given that we intuitively sought event-level models covering approximately 1 to 2 seconds, we chose the model constructed with a transition penalty of 62.5 and 166 frames per state for our experiments. The final segmentation using this model contained 27,432 symbols. We used this model as the event layer for two separate scene models: a full LHMM,

Transition Penalty	Initial Frames Per State				
	33	66	100	133	0.6
0	0.6	0.6	0.7	0.6	0.6
62.5	1.6	1.8	1.8	1.7	1.8
125	2.8	2.6	2.9	2.7	2.9
250	5.2	4.6	5.2	4.9	4.9
500	10.6	9.3	10.9	10.0	9.9

Table 2: Average Event Length By Transition Penalty and Frame Allocation

Model Level	Number of States	Transition Penalty		
		0	0.125	0.25
1	16	3.0	5.8	17.6
2	8	3.3	11.5	114.7
3	4	3.9	49.7	737.9
4	2	7.7	61.0	1098.6

Table 3: Average Scene Length By Level and Transition Penalty, SKM Scene Learning

trained with SKM and an HHMM whose structure was induced using MMI. In the SKM case, we again experimented with several different transition penalties; the resulting average scene lengths are shown in Tab. 3. As shown here, the most gradual increase in scene length is found using a transition penalty of 0.125.

To construct our MMI model, we used the same event-level segmentation that we used to construct our SKM model. We then applied the MMI algorithm described above above with the following set of constraints:

1. A merge is legal if:
   (a) The two states being merged are both on the same HHMM level and this level is lower than the current level *or*
   (b) The two states being merged are not on the same HHMM level *and*
      i. The higher-level state is not on the current level *or*
      ii. The higher-level state is on the current level and it has less than three children and the lower-level state is not already a child.
2. A low-level state sequence must appear at least 10 times to be merged into a new higher-level state.
3. We require a minimum of two new merges per level; if no proposed merges lower $\hat{I}(C)$, we accept the merge that minimizes it.

We used these constraints to construct a five-level HHMM which is described in Tab. 4. Note that only 23 scenes were actually learned; this would seem to indicate that there is actually little short-term predictibility in the source corpus.

4.4 Building the Visual User Context Model

We had much less image data than audio data to work with in this research; hence, we were restricted to a very simplistic visual user context model. We trained a single

Model Level	Scenes Learned	$\hat{I}(C)$	Scene Length
0	0	0.163	1.80
1	16	0.138	2.04
2	2	0.138	2.05
3	3	0.138	2.06
4	2	0.139	2.07

Table 4: Scenes Learned, $\hat{I}(C)$, and Average Scene Length, MMI Scene Learning

Test Subcorpus	Miss Rates		
	False Interrupt	False Reject	Total Miss
1	11.5%	0.0%	11.5%
2	18.2%	0.0%	18.2%
3	5.4%	0.0%	5.4%
Average	11.4%	0.0%	11.4%

Table 5: Miss Rates by Time: Priors Only

diagonal covariance Gaussian model for each interruptibility class using the visual features described in Sec. 4.2.

4.5 Results - Acoustic Information

To establish a baseline, we first tested an approach in which we simply always guess the most fequent class in the training data. In each of the three round-robin experiments, this baseline approach amounts to always guessing that the user is interruptible. The results of employing this approach are shown in Tab 5. This table shows false interrupt rate (i.e. the rate at which the system incorrectly hypothesized that the user was interruptible), false reject rate (i.e. the rate at which the system incorrectly hypothesized that the user was uninterrubtiple), and total miss rate on a per-second basis.

As with the baseline, we tested HMM-based performance using a round-robin approach. Here, we trained the interruptibility likelihood model model $P(S|I)$ sing the training subcorpora, and computed results on the testing subcorpora. We first tested the event-level HMM trained using the SKM algorithm. Miss rates for this model are shown in Tab 6.

The overall miss rate of 8% represents a 30% reduction in error relative to the prior. Further, the false interrupt rate and false reject rate are balanced, which in the absence of specific user preference to the contrary, is preferable to an unbalanced performance profile.

We next tested performance for the multilevel scene models trained with both SKM and MMI. We first computed the mutual information between model states and interruptibil-

Test Subcorpus	Miss Rates		
	False Interrupt	False Reject	Total Miss
1	9.2%	20.5%	29.7%
2	1.2%	1.2%	2.4%
3	4.8%	0.5%	5.3%
Average	4.1%	3.9%	8.0%

Table 6: Miss Rates by Time: Event-level HMM

Learning Method	Model Level				
	0	1	2	3	4
SKM	0.260	0.129	0.195	0.178	0.093
MMI	0.260	0.203	0.204	0.199	0.200

Table 7: Mutual Information Between Model State and Interruptibility

Model Level	Miss Rates		
	False Interrupt	False Reject	Total Miss
0	4.1%	3.9%	8.0%
1	5.9%	26.9%	32.8%
2	19.4%	17.6%	37.0%
3	2.7%	20.7%	23.4%
4	19.3%	19.4%	38.7%

Table 8: Average Miss Rates by Time: Multi-level SKM Learning

ity labels; intuitively, a higher association between state and interruptibility should lead to lower miss rates. Mutual information figures are shown in Tab. 7; in both approaches mutual information degrages on average as levels are added, but the degradation is much more severe in the SKM-trained model.

Miss rates for the SKM approach and the MMI approach are shown in Tabs. 8 and 9. On average, miss rates worsen at higher levels for the SKM-trained model and improve slightly at higher levels for the MMI-trained model; the best result is obtained by the level 4 MMI-trained model. The MMI miss rate of 6.5% represents an 18% relative improvement over the event-level model and 43% over the prior. Although the absolute improvement over the event-level model is small, there is a clear trend, and in any case performance is superior to the SKM model. A likely reason for this performance is that the SKM model capacity shrinks at higher levels, meaning that granularity in the interruptibility likelihood model $P(S|I)$ is lost. Conversely, MMI is extremely selective about which models it merges. As a result, the level 4 MMI model has up to 55 symbols, while the level 4 SKM model has only 2.

4.6 Results - Visual Information

After conducting experiments using audio information only, we considered the addition of image information. The use of image information is natural; many smartphones have onboard cameras with relevant APIs exposed. Further, the

Model Level	Miss Rates		
	False Interrupt	False Reject	Total Miss
0	4.1%	3.9%	8.0%
1	4.1%	3.9%	8.0%
2	4.1%	3.9%	8.0%
3	3.1%	3.9%	7.0%
4	2.9%	3.6%	6.5%

Table 9: Average Miss Rates by Time: Multi-level MMI Learning

Test Subcorpus	Miss Rates		
	False Interrupt	False Reject	Total Miss
1	0.7%	66.7%	67.5%
2	52.7%	6.1%	58.8%
3	0.2%	38.0%	38.3%
Average	28.4%	24.4%	52.7%

Table 10: Average Miss Rates by Time: Image-based GMMs

data collector labelled the corpus for interruptibility using the images as reminders activity; in many cases, visual information can be enough to determine user state. Finally, in many applications, the failure modes of audio and video can be complementary. We thus trained and tested image interruptibility models using the same round-robin procedure that we used for audio. Results are shown in Tab. 10. The overall miss rate of 52.7audio-based miss rates. Given the overall poor performance of the visual subsystem on this corpus, we elected not to pursue a joint audio-visual model at this time. As we discuss in the Conclusions, however, we suspect that lack of data was a serious problem for the image models and we plan to return to this exploration.

5. ADAPTING THE MODEL TO SPECIFIC USERS AND NEW CONDITIONS

One issue with user context modeling approaches of this type is adaptation. It is for the most part impossible to collect enough data to achieve high performance across a large body of users. This is the case because users vary along two dimensions: the types of activities they engage in and environments they visit, and their own interruptibility preferences. Our system offers several advantages in this area over systems such as the one proposed by Siewiorek et. al. in [19]. Specifically, adaptation to novel activities and environments can be achieved by adding new event-level models to the HHMM whenever the best-matching event-level model is a sufficiently poor match to the data. High-level models involving new low-level models can be learned in the same way that the model was initially trained and infrequently used models at all levels pruned, perhaps using an approach similar to the one suggested by Pfleger in [15]. Adaptation to user preference can be achieved by allowing the user to supply negative feedback to the system when it performs incorrectly. In case the system interrupts the user when he considers himself uninterruptible, this feedback is easy to elicit and track; the user presses some form of rejection button. In the opposite case, feedback is somewhat harder to come by; there is no urgency to silence the telephone as there is in the unwanted interruption case. However, since it seems intuitively correct that unwanted interruptions are on average worse than unnecessary rejections, we assume that users of such a system will be cooperative. One issue to address is how to use the feedback. In a deployed system, the sensory interruptibility model we have described is but one module; social models also come into play. It will thus be the responsibility of the main control module to assign blame to the sensory or social module. When blame is assigned to the sensory module, we must specify whether it was an error in the auditory subsystem or the visual subsystem. Once blame is assigned, however, there are two ways

to proceed. First, we can simply update the interruptibility likelihoods $P(S|I)$. An alternate approach involves determining whether or not the sensory data with a given Viterbi segmentation yielding an incorrect interruptibility likelihood is similar to an alternate segmentation which would yield a correct interruptibility likelihood. If this were found to be the case, we could apply discriminative training to make the alternate segmentation more likely in the future. Ongoing research indicates that the first approach, at least, is viable given a scene model with sufficiently broad acoustic coverage, provided that the interruptibility likelihoods are uniform to start.

6. CONCLUSIONS

We have demonstrated a method for estimating user interruptibility from ambient audio and images for use in a smart mobile telephone appplication. We described an interruptibility model based on acoustic information, and showed how to induce HHMM model structure from data using an MMI criterion. We demonstrated, using nearly 14 hours of real-world data, that this approach was effective in learning how to estimate interruptibility. We also evaluated the use of image data to estimate interruptibility and found the evidence for its utility negative, but wholly equivocal due to lack of data. Finally, we described the need for adaptation to new activities, new environments, and user preference. To address this need, we discussed how to elicit user feedback, and how to adapt the model in the face of such feedback.

The work we described here has, in our view, three main shortcomings. First, the audio data we collected was of high quality and required the use of an external recording device. This scenario does not match any truly deployable system; we are currently exploring the recording and use of low-quality audio using only the microphones available on a typical smartphone. Second, the database we used is not large enough to embody a significantly wide range of typical experiences, even for the one user who collected data. Third, the image dataset is extremely impoverished. Images were originally intended to be used only as labeling tools; however, when it became clear that certain visual scenes were extremely suggestive of user state, we began to use them for inference as well. We plan to address both of these shortcomings by collecting more data in a larger variety of settings while capturing images from the mobile telephone camera at much shorter, preset intervals (e.g. 20 seconds). In this way, we will vastly increase the amount of images available for modeling, and also produce higher-resolution interruptibility labels for future research.

7. ACKNOWLEDGMENTS

This research is supported by the European Commission CHIL project (`http://chil.server.de`) under contract No. 506909 and by the National Science Foundation CareMedia project under contract No. IIS-0205219. We wish to thank Maria Danninger, Gopi Flaherty, and the rest of the Connector team at Karlsruhe University and Carnegie Mellon University for their insights and support. We additionally wish to thank the anonymous reviewers of this paper for their useful comments.

8. REFERENCES

[1] B. Clarkson and A. Pentland. Extracting context from environmental audio. In *Proceedings of the 2nd International Symposium on Wearable Computers*, 1998.

[2] B. Clarkson and A. Pentland. Unsupervised clustering of ambulatory audio and video. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[3] M. Danninger, G. Flaherty, K. Bernadin, H. Enekel, T. Kohler, R. Malkin, R. Stiefelhagen, and A. Waibel. The Connector — facilitating context-aware communication. In *Proceedings of the International Conference on Multimodal Interfaces*, 2005.

[4] M. Danninger, T. Kluge, E. Robles, L. Takayama, Q. Wang, R. Stiefelhagen, C. Nass, and A. Waibel. The Connector service — predicting availability in mobile contexts. In *Proceedings of the Joint Workshop on Machine Learning and Multimodal Interfaces*, 2006.

[5] D. Ellis and K. Lee. Features for segmenting and classifying long-duration recordings of personal audio. In *Workshop on Statistical and Perceptual Audio Processing*, 2004.

[6] D. Ellis and K. Lee. Minimal-impact audio-based personal archives. In *First ACM Workshop on Continuous Archiving and Recording of Personal Experiences*, 2004.

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

[8] E. Horvitz and J. Apacible. Learning and reasoning about interruption. In *Proceedings of the ACM International Conference on Multimodal Interfaces*, 2003.

[9] S. Hudson, J. Fogarty, C. Atkeson, D. Avrahami, J. Forlizzi, S. Keisler, J. Lee, and J. Yang. Predicting human interruptibility with sensors: A wizard of oz feasibility study. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2003.

[10] R. Malkin and A. Waibel. Classifying user environment for mobile applications using linear autoencoding of ambient audio. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[11] R. Malkin and A. Waibel. The 2006 CMU CLEAR acoustic environment detection system. In *Proceedings of the 2006 CLEAR Evaluation Workshop*, 2006.

[12] K. Murphy. Hierarchical hmms. Technical report, Computer Science Department, University of California at Berkeley, 2002.

[13] K. Murphy and M. Paskin. Linear time inference in hierarchical HMMs. In *Proceedings of Neural Information Processing Systems*, 2001.

[14] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Proceedings of the International Conference on Multimodal Interfaces*, 2002.

[15] K. Pfleger. *On-Line Learning of Predictive Compositional Hierarchies*. PhD thesis, Stanford University, 2002.

[16] M. Reyes-Gomez and D. Ellis. Selection, parameter estimation, and discriminative training of hidden Markov models for general audio modeling. In *Proceedings of the International Conference on Multimedia and Expo*, 2003.

[17] K. Ries, F. Bub, and Y. Wang. Improved language modeling by unsupervised acquisition of structure. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[18] D. Ron, Y. Singer, and N. Tishby. Learning probabalistic automata with variable memory length. In *Proceedings of the International Conference on Computational Learning Theory*, 1994.

[19] D. Siewiorek. Sensay: A context-aware mobile phone. In *Proceedings of the International Symposium on Wearable Computers*, 2003.

[20] M. Slaney. Semantic-audio retrieval. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[21] L. Xie, S. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. In *Proceedings of the International Conference on Multimedia and Expo*, 2003.

[22] L. Xie, S. Chang, A. Divakaran, and H. Sun. Learning hierarchical hidden Markov models for video structure discovery. Technical report, Department of Electrical Engineering, Columbia University, 2004.