Machine Listening for Context-Aware Computing

Robert G. Malkin

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

December 14, 2006

Thesis Committee:
Dr. Alex Waibel, Chair, LTI
Dr. Michael Lewicki, CNBC
Dr. Richard Stern, ECE/LTI
Dr. Jie Yang, HCII/LTI/

Abstract

Machine listening is an area of study which is rapidly increasing in importance. The proliferation of massive sensory corpora, together with the perceptual needs of smart computational devices and smart spaces has lead to this increase. Machine listening provides both a computationally cheap alternative to machine vision, and a source of information that is complementary to visual information; hence, perceptual systems which lack the ability to process auditory information will in general perform less well than those which can process auditory information. Machine listening is also interesting in its own right, as research into computational auditory processing can help to shed light on general principles of perception, and on how our own perceptual systems work. This thesis describes machine listening research designed to solve real-world problems in perceptual and context-aware computing.

This thesis makes two claims. First, it claims that machine listening technologies are well-suited to the task of providing context awareness in real-world computational systems, whether these systems are intended to provide operational cues to smart devices or spaces, or to segment, summarize, or select segments of interest in multimedia corpora to make them more useful to human users. Second, it claims that the use of the core principle of

perception, redundancy reduction, can guide the design of practical systems to provide context awareness in this way. The validity of these claims is supported by evidence from three application areas: multimedia gisting, acoustic environment recognition, and estimation of user interruptibility for the CHIL Connector service, a smart mobile telephone.

Acknowledgments

The acknowledgments page is typically the place where prospective Ph.D.s aver in the strongest terms that any aspirations they had of finishing their degrees would never have been realized without the help and sacrifices — or at least the complicity — of others. And so here I mention these others by name, heaping praise upon them as I go.

I first thank my advisor, Alex Waibel, for everything I have accomplished at CMU, including this thesis. Alex had the foresight to encourage research into the relatively unexplored field of non-speech acoustic analysis, and provided the guidance and first-rate research environment required to make it a success. I also thank the rest of my committee; Mike Lewicki, Rich Stern, and Jie Yang, for providing the support and advice I needed to finish this work.

The “sacrifices” portion of this acknowledgment comes next. I cannot thank my wife Katya enough for supporting me in this work for lo these many years. It has become cliche to state that one’s thesis is as much one’s spouse’s as one’s own, but it is largely true. It also belongs to my daughters, Emma Jane and Hannah Elizabeth. And I suppose also to the Malkin-to-be-named-later.

I am of course indebted to the rest of my family for all they have done for me. My

parents, Gary and Rosemary, gave me and my brother David a loving home in which to grow and learn. My grandparents Walt, Mary, W.K., and Sylvia did more for me than I am aware, and I am saddened that only one of them will get to read this. Katya's mother Cindy also deserves a spot in these acknowledgments for all she has done to help us with the kids; this help has made a huge difference.

I mentioned earlier the quality of the research environment that Alex has created; I thank all of my colleagues at the Interactive Systems Laboratories at CMU and UKA for their friendship, collaboration, and support. Those who are deserving of specific mention include Sondra Ahlen, Markus Baur, Michael Bett, Susi Burger, Maria Danninger, Matthias Denecke, Michael Finke, Gopi Flaherty, Isaac Harris, Denise Hill, Florian Kraft, Kornel Laskowski, Thomas Schaaf, Tanja Schultz, Rainer Stiefelhagen, Minh Tue Vo, Hua Yu, and Torsten Zeppenfeld. Kornel, Isaac, and Denise in particular get extra praise for their efforts to collect interesting databases for me to work with. Isaac also gets special notice for taking on the position of systems administrator here at interACT and sparing me the agony of dealing with yet another stale NFS filehandle. Datong Chen, Andrey Temko, and Christian Zieger also deserve mention as valued research collaborators.

Before I was a graduate student at CMU, I had to survive being an undergraduate at CMU. Eight people were instrumental in ensuring that I did: Mike Andrew, Alan Horn, Jeff Jesensky, Ron Masztak, Patrick McGrath, Alex Rutkowski, Dave Saylor, and Trevor Smith. Completely unironic thanks also must go to the national office of the ATΩ fraternity for revoking our charter.

Finally, there are the people who, without conscious intent, helped make this thesis possible by influencing my earlier life decisions. Rich Lackner, the CMU head football coach, convinced me to enroll here instead of at the University of Chicago. His advice in this matter was absolutely correct, and so in a sense this whole thing is his fault. Two high school teachers of mine, Bob Trifone and Joe Schneirlein, had an immense effect on my early development, for which I am very grateful. Last and certainly close to least,

Eugene DiPasquale, the proprietor of the Panther Hollow Inn, also influenced my career in a number of interesting ways.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Overview

Machine listening is an area of study which is rapidly increasing in importance. The proliferation of massive sensory corpora, together with the perceptual needs of smart computational devices and smart spaces has lead to this increase. Machine listening provides both a computationally cheap alternative to machine vision, and a source of information that is complementary to visual information; hence, perceptual systems which lack the ability to process auditory information will in general perform less well than those which can process auditory information. Machine listening is also interesting in its own right, as research into computational auditory processing can help to shed light on general principles of perception, and on how our own perceptual systems work. This thesis describes machine listening research designed to solve real-world problems in perceptual computing.

1.2 Motivation

Massive corpora of multimedia data are every day becoming cheaper and easier to create and store. Capture and storage devices, once expensive, bulky, and of limited capacity, are now inexpensive, lightweight, and can store scores of hours of audiovisual data. As a result of this continuing trend, large multimedia corpora consisting entirely of *spontaneous* data — that is, data which is largely unscripted, as opposed to film, sporting events, or broadcast news — are rapidly proliferating. The sources and potential uses of these spontaneous corpora are many. They can be produced by individuals wishing to capture personal experiences to construct multimedia diaries. They can be produced by humanoid robots learning to interact with humans in real-world situations. They can be produced by surveillance apparatus to monitor patients in hospitals or nursing homes, or to enhance the security of an indoor or outdoor space. They can be produced by smart spaces or devices learning how to respond to the needs of humans who use them. They can be produced by individuals engaged in team activities such as search and rescue or military operations, and used as after-action review or training material. Finally, they can be produced by intelligence-gathering apparatus and used to build or maintain awareness of geopolitical situations.

Several common threads join all of these types of multimedia corpora. First, they are spontaneous. This means that they are unscripted and lack patently obvious state-change cues like blank screens. Second, they are often sparse. This means that the vast majority of spontaneous data is uninteresting for any particular purpose. Third, they are large. Coupled with their sparseness, this means that it is difficult for humans or machines to find what they are looking for. This difficulty places a hard limit on the utility of large, spontaneous multimedia databases. In order to fulfill the promise of these corpora, some automatic means of indexing and querying them is needed.

The problem of indexing these corpora is cast as a problem of scene analysis. That is, we can attempt to find certain features of the corpus which serve as concise explanations of

the data. These features include scene boundaries, multimedia summaries or gists, labels describing specific events and environments, and multiresolution structures. Each of these features serve a slightly different purpose, but all are steps toward making large multimedia corpora more useful.

For human users, scene boundaries can be used to find areas of large-scale change, which are in theory the most informative regions of the corpora. Multimedia summaries help users explore the main actions of the corpus, determine whether or not it contains anything of interest, and quickly construct a story for personal consumption of sharing. Environment and event labels allow users to quickly find specific areas of interest, if they already know what they are looking for. Finally, multiresolution structures help users quickly explore corpora in detail, first by browsing at low levels of detail, and then drilling down into areas of interest.

Machine users of these corpora can also benefit from these types of analysis. Scene boundaries and summaries can help robots and smart spaces to identify those regions which, when deeply analyzed, will yield the most information about their surroundings. Environment and event tags can help robots, smart spaces, and smart devices to understand the social context of the humans they are trying to serve, and to make better decisions about how to interact with them. Finally, multiresolution structures can help robots, smart spaces, and smart devices to predict future events and scenes, enabling them to anticipate rather than just react to the needs of the humans they are trying to serve.

1.3 Machine Listening for Context Awareness

In this thesis, “machine listening” is defined as the process of deriving from the audio signal information that is useful for some computational or human purpose, examples of which are described above. Analysis of the audio signal is well-suited to these kinds of context awareness problems for many reasons. First, the signal is relatively compact. This means that

storage and and computation are relatively cheap compared to video [1]. Broadcast-quality audio only takes up 109 megabytes per hour, and audio features typically used for high-resolution analysis typically use less than 18 megabytes per hour. Further, the processing required to derive these features from the raw signal, based on the fast Fourier transform (FFT), is very inexpensive. Second, sensors are for the most part cheap and robust. Quality stereo microphones can be had for less than $100 and are highly resistant to damage from rough handling. Third, omnidirectional microphones pick up sound from all directions, meaning that the signals they record are unaffected by orientation. One does not have to rely on exact sensor placement, or ask humans to alter their natural behavior in order to use onboard audio sensors. Fourth, the audio signal is impervious to changes in lighting conditions. Fifth, the audio signal is not affected by occlusion in a perceptually relevant way; an audio signal can pass through walls and around obstacles and for the most part retain its perceptually relevant features. Sixth, and perhaps most important, events that occur in the real world which are relevant context cues very often leave behind acoustic evidence which can be identified by humans and discovered with techniques from pattern recognition and machine learning. Finally, the audio signal is often complementary, if not orthogonal, to the video signal, which has been extensively used for context awareness. Adding audio to video-based systems should be a cheap way to improve performance or speed or both. See for instance the work of Chen et. al. [12], [13], in which audiovisual data collected in a nursing home environment is first processed with audio event detection techniques and only subsequently by video event classification techniques. This work demonstrates that cheaper audio processing can save a significant amount of video processing for certain tasks.

It should be emphasized again, though that this is not the only purpose of machine listening. Indeed, there are many context awareness applications where audio is the best

[1] Though recent results in machine vision suggest that certain high-level image analysis procedures can be performed quite cheaply; see for example work by Torralba and Oliva [103].

modality, and vision, if appropriate for use at all, plays a complementary role. Perhaps the best example of such an application is automatic speech recognition (ASR), which yields detailed context information from audio, and only under certain conditions can be helped by visual analysis (i.e., lipreading in adverse noise conditions).

Taken individually, these are all good arguments for conducting research into using auditory information to inform context-aware computing systems. Taken together, they constitute a very powerful argument for such research. This thesis helps to validate these arguments by giving multiple examples of applications deriving context awareness from audio.

1.3.1 Machine Listening Theory

The theory of machine listening is largely the theory of perception, adapted to the special demands of processing the audio signal. The underlying mechanism of perception in biological systems is by now well-established. Organisms are exposed to a highly redundant input signal with high dynamic range in real time. After compressing this signal into some smaller dynamic range via saturation, they analyze the signal's contents in order to achieve environmental awareness and plan appropriate actions.

The analysis step is the focus of perceptual theory. The analysis step can be viewed as a decomposition into feature extraction and pattern recognition steps. In computational systems this decomposition is literal, and very different procedures are used for each step. In organic systems, though, the distinction is much less clear. Organic systems process the input signal, find specific patterns, and produce new, lower resolution signals, which are in turn used by higher levels of processing. Much of the processing done by organic systems is concerned with finding and eliminating redundancies, or coincidences that are present in the input signal. The act of redundancy reduction, which amounts to signal compression, is widely viewed to be the essence of perception. Speech recognition is an excellent example

of this phenomenon; digital broadcast-quality audio has a bit rate of 62.5 kb per second, yet the total amount of information present in one second of speech is much less than this figure: two or three bytes worth of words, and perhaps a few bytes worth of meta-information dealing with speaker identity, emotional state, and similar features. Humans perform this compression with the greatest of ease; computational systems designed to perform similar tasks are very complex and computationally expensive.

Listening presents special challenges to a perceptual system based on redundancy reduction. First, the audio signal retains perceptual invariance under environmental conditions that can radically alter the observed values of the signal. The signal is altered by variations of the medium of transmission, and by the shape and material of objects in the path between source and listener, and the resulting amplitudes can be quite different depending on the exact conditions. The fact that perceptual invariance holds under these conditions means that the redundancies exploited by the auditory system are independent of the actual frequencies or amplitudes in the audio signal, and they must instead relate to the relationships *between* frequencies and amplitudes. Second, the audio signal is one-dimensional and changes very rapidly. This means that, unlike the visual domain, the listener never has a complete picture of a sensory object at any particular instant. Instead, the listener must maintain a sensory buffer over which to find redundancies; in other words, redundancies in the audio signal are temporal rather than spatial redundancies (though the audio *spectrum* contains temporal, spatial, and spatiotemporal redundancies). Third, the audio signal is factorial in nature. The signal reaching the listener is the sum of many signals emanating independently from many sources, all modified by the medium of transmission, environmental objects, and each other. Organic systems are able to separate these sources, or at least to extract one source at a time from the din for analysis, a process exemplified by the cocktail party effect. This ability to separate a single source from a mixture without additional cues implies that the relevant redundancies are characteristic of one source or another, and can be identified and tracked over time.

The principle of redundancy reduction applies to organic listeners in a straightforward way; the neural coding apparatus seeks ever-lower resolution representations of the input signal and stores, uses, or forgets them as appropriate. The mapping of redundancy reduction to machine listeners is somewhat less clear. Engineering realities mean that one often does not seek to implement listening systems which are perfect models of the mammalian auditory periphery and cortex; different types of models are typically used for each stage of processing called for by the application: feature extraction, acoustic modeling, and application of prior knowledge all are handled separately. Nonetheless, the principle of redundancy reduction can be a useful guide to the engineering choices one makes when building real-world listening applications.

1.3.2 Applications of Machine Listening

Applications of machine listening that provide context awareness to smart devices, spaces, and media archives are compelling both for theoretical reasons and because the applications they permit are of value to users. In this thesis, the focus is on three such applications. First is user state modeling for mobile devices and smart spaces, which is cast as a problem of acoustic environment recognition. Second is sensory gisting and report generation for multimedia skimming, mining, and summarization, which is cast as a problem of acoustic blind value assignment. Third is activity recognition and prediction, which is cast as a problem of learning and identification of multiresolution structures using acoustic information. These applications have some overlap in that techniques which are useful for one area could sometimes useful in another; for example, the relationship between user environment and user activities can be strong. It is logically expedient, however, to present them separately, as they make use of different techniques of implementation and evaluation. These technologies are now introduced in turn.

User State and Environment Modeling

User state encompasses any number of factors that can influence what demands a human user makes of a computational system, and how the user expects these demands to be fulfilled. In well-defined, enclosed spaces like offices, specific events can be detected which serve as important context cues. In mobile applications, however, there is a potentially enormous set of events which could yield context information. Further, acoustic conditions can vary drastically from locale to locale, making specific events hard to detect. These two features of the mobile application space present an opportunity as well as a problem. While it may be difficult to detect specific events in a mobile setting, it should not in principle be difficult to detect what kind of environment the user is in based on auditory data. It seems intuitively plausible that certain types of sound fields are characteristic of certain environment types. For instance, the sound field of a city street will be dominated by turbulent noise from automobiles, while the sound field of an office will be dominated by speech, typing, and other typical office sounds. The overall dryness or reverberance of the sound field can be an important cue as well. If, as intuition suggests, environments do have characteristic sound fields, then environment detection from audio should be possible.

The two most compelling uses of environment modeling are for mobile devices and autonomous robots. Mobile devices are often used by humans to gather information; as wireless broadband becomes more prevalent, the number of humans using PDAs or smartphones to seek information over the world wide web in mobile settings will increase. In order to improve latency, environment information could in principle be used to predict user behavior and pre-fetch relevant information. This process could be performed with simplistic common-sense relationships; for instance, a PDA which can detect that its user is in an airport might seek and pre-cache updated flight information. It could also be performed by learned, user-specific preferences; for instance, a PDA might learn that its user often seeks stock quotes on a morning bus ride. Autonomous robots could use environ-

ment information to help maintain an awareness of appropriate behavior independent of actual locale (as reported, for instance, by a GPS receiver). Some of this information could be obtained through visual processing, as one would assume that a mobile robot, unlike a PDA, is equipped with appropriate visual sensors. In this case, audition makes sense as a complementary sensor modality, as there may be many environments which are confusable in one modality but not the other.

In this thesis, environment recognition is viewed as a gestalt process. That is, no attempt is made to explicitly detect or model known characteristic events; rather, a self-organizing model is selected and learned for each environment. Note that environment recognition is not the same problem as locale recognition; the system presented in this thesis seeks to achieve generality where locale recognition systems simply seek to learn the characteristic sound fields of specific locations.

Multimedia Sensory Gisting

Multimedia sensory gisting is the process by which streams of multimedia data are summarized by selecting sets of segments of those streams in such a way that the selected segments contain an overview, or the gist, of the activities represented therein. This procedure amounts to automatically deciding which segments in a stream are the most interesting, for some possibly application-dependent definition of interesting. Applications of this type generally fall into two categories: knowledge extraction, which is inherently an offline process, and remote aggregate sensing, which is inherently an online process. While it is possible to pursue approaches to these problems based on semantic analysis of the source data; i.e., approaches in which certain classes of events are known *a priori* to be interesting, the same difficulties hold as for user state modeling. The size of the event space is possibly unbounded, and unstable acoustic conditions may make it difficult to detect familiar sounds in unfamiliar environments. For these reasons, this thesis explores the use of *blind*

techniques which make no *a priori* assumptions regarding the contents of the signals under analysis, either in terms of environments or events. Blind techniques thus make heavy use of analysis of the statistics of the signals in a manner consistent with the principles of perception.

Knowledge extraction encompasses many different kinds of applications, including personal experience distillation and storytelling, surveillance, data mining, and after-action report generation. The gists created for these kinds of applications can take one of two forms. The first form emphasizes detection and presentation of scene changes, which often involve a marked change in acoustic texture in such a way that the audio is heterogeneous across the change and homogeneous on either side of the change. There are theoretical reasons to prefer this kind of summary. Scene changes are those points in the signal where redundancy breaks down; in other words, they represent new information which can be seen as by definition interesting. An algorithm for producing this form of gist might seek the n strongest scene changes, where n is the desired number of segments in the gist. A second form emphasizes short-term events (which can be seen as scene changes over a short timescale). In this form of gist, a collection of segments is selected such that their contents are maximally mutually dissimilar, irrespective of whether or not they contain a large scene change.

Remote aggregate sensing encompasses applications involving teams of individuals cooperating to perform some task, either collocated or not and possibly coordinated by some central authority. Applications of this type include firefighting, search and rescue, mulitparty gaming, and other multiparty tactical scenarios. Here, the participants and planning authority would like to have real-time awareness of the states of the entire group in order to better inform their decision-making processes. Verbal communication in these problem spaces may be impractical either for safety reasons or because of the possibility of channel saturation or noise. Broadcasting of multimedia data captured from wearable sensors is a better solution. However, this solution has the potential to cause cognitive overload; the

more real-time streams are added to the broadcast, the harder it becomes to know which ones merit attention. This problem is quite similar to the offline knowledge extraction problem discussed above. Here, though, the goal is to rank in real time each signal according to how much attention it merits, or equivalently, how interesting it is.

Multiresolution Scene Identification and Structure Learning

As noted above, acoustic events and environments can offer important clues to user state. In stationary environments with known usage patterns, a more or less complete set of acoustic events can be identified and used as state cues. In smart spaces intended to support some activity, like meetings, presentations, or daily work routines, these events can help identify what kind of activity is currently taking place. The goal is to use this information to *anticipate* the needs of the humans in the space and thus spare them the inconvenience of having to ask. For example, at the beginning of a meeting, a smart space might automatically show an attendance list; at the end, it might display automatically-generated minutes or action items. Context cues that are useful for identifying these kinds of states can be related to vision and speech; human identities can be visually confirmed, and humans often talk about what they are doing as they are doing it. However, there are some cues that are both auditory and non-verbal, and machine listening can help to detect these. These kinds of cues include events like door slams, telephone rings, applause, typing, footsteps, electrical noises, presence or absence of speech, babble noise, and others. Many of these events are the result of actions that have important implications regarding the state of the space and thus the kinds of requests the space should expect.

However, simply using point events as state-change triggers may be problematic in systems intended for long-term use, such as an always-on smart space, a personal diary system, or a smart mobile device. Here, in addition to point events, longer-term activities and patterns of activities are also of interest. These patterns, or scenes, can be used in three ways.

First, knowledge of high-level scene context can help guide expectations of what events are likely or unlikely to occur; this knowledge could in principle improve event detection performance. Second, having a high-level scene label for each instant in a sensory record might make it easier to browse and extract knowledge from the record, either manually or automatically. If multiresolution scene labels are available, human or machine consumers of the sensory record can browse at low levels of temporal detail and drill down into areas of interest. Third, high-level representations might improve context-dependent performance for specific applications; for example, a context-aware mobile telephone like the CHIL Connector [25] might be better able to make decisions about its user's interruptibility given a high-level activity representation than given a low-level representation.

In this thesis, multiresolution scene learning is seen as a bottom-up, unsupervised clustering problem. Structures, or scenes, are defined in terms of compositional relationships. That is, scenes are composed of more or less well-defined sequences or groups of events which can themselves be part of longer and larger patterns. In an office application, for example, there are many patterns of activity which happen on a number of timescales. A typical workday in an office setting, for example, might consist of five main activity cycles; arrival, quiet morning work, lunch, afternoon meetings, and departure. Each of these activities can itself be broken down into smaller subscenes and so on down to atomic events like opening the door or typing on a keyboard. The typical workday is also part of larger patterns, like the 5-day workweek or annual activity cycles. The ability to identify these short and long patterns is a potentially tremendous source of information for context aware settings.

The work in this thesis focuses on learning scenes from unlabeled data in a mobile setting, and using these scenes to perform a specific context-awareness task; namely, detecting whether or not the user of a smart mobile device is interruptible in the sense of being able to accept an incoming phone call independent of the identity of the caller. An approach to scene learning based on redundancy reduction is presented and used to build

an interruptibility detection system.

1.4 Thesis Statement

This thesis makes two claims. First, it claims that machine listening technologies are well-suited to the task of providing context awareness in real-world computational systems, whether these systems are intended to provide operational cues to smart devices or spaces, or to segment or summarize multimedia data in order to make them more useful to human users. Second, it claims that by adapting general principles of perception to the auditory domain, practical systems can be built which provide this kind of context awareness. Evidence for these claims is provided by measuring the performance of these systems on real-world sensory data for three specific tasks: gisting of multimedia data, environment recognition, and scene learning for mobile interruptibility estimation.

1.5 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2, *Machine Listening Theory*, provides the relevant theoretical background on machine listening, audio processing, and machine perception in general. It also contains a brief review of hidden Markov models.

Chapter 3, *Auditory Gisting for Summarization and Stream-of-Interest Selection in Multimedia*, shows how very simple perceptually motivated measures of value can be used to perform gist-creation tasks in multimedia data using audio only. The resulting value judgments are shown to be broadly similar to human value judgments.

Chapter 4, *Auditory Environment Recognition for User State Modeling,* shows how user environment can be inferred automatically from the audio signal in a way that could be exploited by mobile context-aware applications.

Chapter 5, *Auditory Scene Learning for Multiresolution Context Awareness,* shows how hierarchical structures can be inferred from audio data in an unsupervised manner.

Chapter 6, *Conclusions,* summarizes the important contributions of this thesis and sets out a plan for future exploration.

Chapter 2

Machine Listening Theory

2.1 Chapter Overview

This chapter contains the bulk of the background required to understand the material in this thesis. It begins with a discussion on the nature of the audio signal (mostly drawn from Yost [115]) and basic audio analysis before presenting relevant results from theoretical machine perception in general and machine audition in particular. The chapter concludes with background material on feature extraction and time series modeling.

2.2 The Audio Signal

The signal that listeners experience as sound is caused by the vibrations of objects in the environment. These vibrations are propagated through a medium, typically air, as waves. The distance between peaks in the wave function is called the wavelength of a sound and is inversely proportional to the sound's frequency. The amplitude of the wave affects the

pressure generated by the wave and is related to perceptual loudness.

As a sound wave propagates outward from its source, it is modified in several ways. First, the amplitude decays with the square of the distance traveled from the source, limiting the effective range of a sound. This decay is dependent upon properties of the medium of transmission. Second, sound waves can encounter objects in the environment. These objects, depending on their shape, composition, and size relative to the wavelength of the sound, can to varying degrees absorb, transmit, or reflect the sound.

When the interfering object is large compared to the wavelength of the sound, the sound is reflected. When the wave reflects back on itself, there are points where two peaks collide, and points where a peak will collide with a trough. When peaks collide, the result is constructive interference; i.e., the sound at that point will increase in amplitude. When a peak collides with a trough, the result is destructive interference; i.e., the sound at that point will decrease in amplitude. Constructive and destructive interference are not limited to the case of sound reflection; different sounds can collide in this way and interfere with each other.

When the interfering object is at most as large as the wavelength of the sound, different effects can occur. If the object is much smaller than the wavelength of the sound, the sound wave will pass the object largely unmodified. If the object is nearly as large as the wavelength of the sound, a sound shadow will be created behind the object whose shape and size depend on the shape and size of the object. In this shadow, the sound is inaudible; beyond this shadow the sound is audible This process is similar to visual occlusion; in practice, however, it is much less of a barrier to omnidirectional sensory awareness.

The summation of all the sound sources, objects, and sound waves in a given environment interacting with each other in complex ways, is called a sound field. It is the sources in this field and the field itself that are of interest in this thesis.

2.3 Audio Signal Analysis

The signal that reaches a biological or mechanical listener is, as stated above, the sum of all the sound waves produced by all the sources in the listener's sound field, possibly modified by the medium and objects in the environment, including the listener. We now address how a listener can turn this signal into useful information. As acoustic energy reaches the listener, it is absorbed and transduced by a membrane (i.e., the eardrum) into mechanical energy. This mechanical energy is further transduced into electrical energy whose intensity is proportional to the intensity of the acoustic signal.

In digital listeners, the continuous amplitude stream must be discretized. This process is called analog-to-digital conversion, or ADC. According to the Nyquist theorem, the signal must be sampled at a rate twice the highest frequency that we are interested in analyzing to avoid aliasing. The discretized audio signal is referred to as the waveform or the time signal. The time signal is a complex function which is not amenable to easy analysis. However, it is possible to represent any function as a weighted sum of simpler functions which are easier to analyze. In audio signal analysis, it is convenient to work with sinusoidal functions which have fixed frequency; signals represented in this way are said to be in the frequency domain. Representing the time signal as a weighted sum of sinusoids allows us to observe the intensities of all frequencies present in the signal. This collection of intensities is called the power spectrum, and can be computed by means of the fast Fourier transform (FFT). Since the audio signal changes rapidly, and these changes are often semantically relevant, it is additionally necessary to compute not just a single spectrum over a given signal, but many spectra over time. This representation, called the spectrogram, is computed by sliding an analysis window of fixed length, usually referred to as a frame, over the time signal and computing a separate power spectrum for each frame. This process is referred to as the Short Time Fourier Transform (STFT). Choosing the optimal frame size for the STFT involves balancing accuracy in time and in frequency. Long frames result in high

frequency acuity and low temporal acuity; short frames result in low frequency acuity and high temporal acuity. ASR systems typically use frames 20 milliseconds long, overlapping by 10 milliseconds.

Biological listeners perform a similar decomposition of the time-domain signal into a set of simpler, easier-to-analyze functions. Unlike our standard machine approaches, they do *not* seem to use pure sinusoids; rather, they use functions that are specially adapted to optimize the functionality of their listening systems. The way in which they do this, and indeed the way in which all organisms perform all perceptual tasks, is addressed in the following sections.

2.4 Perceptual Theory and Computational Audition

There has been interest in organic perception in the psychological community for many decades. In the last 50 years, a great amount of progress toward quantifying organic perceptual processes has been made by a host of researchers. Here, I give a brief overview of some of this work, followed by an introduction to the basic concepts of redundancy reduction as the organizing principle of both organic and computational perception.

2.4.1 Prior Work in Perceptual Theory and Computational Audition

As described in Paris Smaragdis' Ph.D. thesis [93], much of the early work on perception focused on vision; nonetheless, even the earliest perceptual work by Barlow ([4], [5]), Atteneave ([3]), and others were exploring the now-accepted notion of redundancy reduction as a core principle of perception.

Much early research on audio processing focused on coding speech for telephone transmission [62] and on ASR (see [106], [81], [52]). An unrelated research track, auditory scene analysis (ASA), sought to explicitly model and reproduce the means by which hu-

mans perform stream segregation — that is, the separation and identification of specific sound sources over time in complex sound fields. Bregman's seminal work [10] contains an excellent overview of the main results of psychological research into auditory phenomena. Computational auditory scene analysis (CASA) was an outgrowth of Bregman's work which sought to directly implement many of the heuristics Bregman described. The Ph.D. theses of Cooke [22], Brown [11], and Ellis [28] all successfully implemented source separation in some real-world tasks. Wang, Terman, and Liu implemented an alternate approach [108] based on neural oscillator networks [109].

The similarity of the auditory scene analysis problem to the blind source separation problem led many researchers to again begin to view audition in terms of redundancy reduction. Comon [21] introduced the idea of independent component analysis (ICA), a means of decomposing data into simpler functions in a data-dependent manner such that the new functions were statistically independent. Soon after, Atick [2] proposed (or re-proposed) the idea of using information theory as an organizing principle for sensory processing in general. Other researchers soon began to explore and use ICA for computational models of perception. Bell and Sejnowksi in [6] presented the *infomax* algorithm as a method for addressing problems of blind source separation and deconvolution. Infomax was based on maximizing the joint entropy of the coded feature space. This work was followed by demonstrations of infomax used for basis decomposition of both sounds ([7]) and images ([8]). The work on images was particularly enlightening, as it showed that the optimal image encoding was based on visual edges, long believed to be the most informative regions of images. Hyvärinen and Oja presented a new approach to ICA based on maximizing the non-gaussianity of the coded feature space in [49], and, with Oja, have since demonstrated that maximizing sparsity and temporal coherence leads to similar basis decompositions as maximizing non-gaussianity ([44], [45], [46]). In [47], the provided a framework for video coding which unified independence, temporal coherence, and topography into a single unified model. In [58], Lewicki showed that the optimal set of basis functions for audio were

dependent on the specific kinds of sounds in the sensory input. Specifically, the basis functions learned for non-harmonic environmental sounds were similar to wavelets; they were localized in time but not in frequency. Conversely, the basis functions learned for animal vocalizations resembled sinusoids similar to Fourier decompositions; they were localized in frequency but not in time. Finally, the basis functions learned for human speech — which contains both harmonic and non-harmonic sounds — were localized in both time and frequency; further, the degree to which they were localized in time was dependent on frequency. The set of basis functions learned from human speech resembled gammatone functions, long used in phsychoacoustically-motivated auditory research. More recently, Lewicki and Smith ([95], [96], [96]) have worked extensively on learning efficient representations of auditory phenomena using spike timing codes. Finally, Smaragdis' Ph.D. thesis, *Redundancy reduction for computational audition, a unifying approach*, [93] focused exclusively on using redundancy reduction as the core principle of audition and significantly, was able to demonstrate how computational systems for grouping and scene analysis / source separation could be built using this theoretical framework.

2.4.2 Redundancy Reduction as an Organizing Principle for Computational Audition

As these and other researchers have demonstrated, redundancy reduction as an organizing principle for perception in general and audition in particular is motivated by computational efficiency and can be observed in biological systems. All organisms which are capable of perception face a daunting task: how to extract information from the environment in such a way that their likelihood of survival is enhanced. The process of information extraction can be thought of as a coding problem: one signal type is represented by an activation of one group of neurons, while another signal type is represented by an activation of some different group of neurons. Each group responds optimally to a certain stimulus, often referred

to as a *basis function*, in signal space. The activations of the many groups of neurons dedicated to sensory coding are used to learn and recognize important stimuli. Making the coding task difficult is the fact that organisms must carry it out with limited dynamic sensor range and limited processing bandwidth compared to the range of natural signals. Limited sensor range means that sensors must be capable of saturating in order to limit the dynamic range of the input signal. Limited bandwidth means that organisms must code the range of natural signals efficiently. Efficiency here means that the coding process must make maximal use of all coding resources, or equivalently, that system throughput must be maximized. Resources that are not used maximally are in essence wasted, a situation to be avoided. Mathematically, for all resources to be used maximally, they must be used with approximately equal frequency. That is, if there are, for example, n binary neurons which can be used to encode events observed in some real-world signal, each neuron must be activated approximately $\frac{1}{n}$ of the time, no two neurons should be active at the same time, and there should be no observable correlations between the activations of one neuron and another.

The reason that this kind of efficiency can be achieved is that natural signals are highly redundant in both time and space. In vision, if a certain point in space (relative to the viewer) and time is a certain color, it is very likely that neighboring points in space will be the same color, and very likely that the point will be the same color in the next instant in time. Thus, it is of vital importance for a coding system to use more resources on locating and representing the visual edges in an image, where these redundancies do not hold. Audition is similar to vision in this regard. Since auditory events have temporal extent, intensities from instant to instant at a given frequency tend to be similar; likewise, there are often correlations between intensities at different frequencies at the same instant. These correlations are sometimes non-local. As with vision, it is thus imperative to locate and represent regions where the redundancies do not hold — onsets, offsets, and disruptions in non-local frequency correlations.

Coding natural signals in the same way that they are received, e.g., pixel-for-pixel in video, or frequency-for-frequency in audio, would thus be inefficient; many neurons will encode information which could have been inferred from neighboring neurons. Organic sensory systems are therefore largely concerned with finding codes which will allow these redundancies to be reduced and efficiency to be achieved. In order to emulate these accomplishments in computational systems, researchers turn to information theory, which provides measures which can help to formalize notions of efficiency and redundancy. These measures make statements about the information content of statistical distributions of random variables; if we view sensory inputs and outputs as random variables, we can use information theory to guide the search for optimal codes.

The most basic measure in information theory is the information I of an event, or specific value, x of a random variable X with respect to some probability distribution P, as shown in Equation 2.1. The information of an event is proportional to the log of the inverse of its probability; the rarer an event, the more information is provided when it occurs. The average information over all events of a random variable is the entropy, denoted $H_P(X)$ and given in Equation 2.2. The closer a distribution is to uniform, the higher the entropy. The maximum-entropy distribution over an N-element set of discrete events is the uniform distribution, whose entropy is $\log N$. The Shannon redundancy, $\mathcal{R}_P(X)$, measures the degree to which the entropy of X differs from the uniform distribution, and is given in Equation 2.3.

$$I_P(x) = \log \frac{1}{P(x)} = -\log P(x). \tag{2.1}$$

$$H_P(X) = -\sum_{x \in X} P(x) \log P(x). \tag{2.2}$$

$$\mathcal{R}_P(X) = 1 - \frac{H_P(X)}{\log N}. \tag{2.3}$$

The Shannon redundancy can be used as an objective function for perceptual coding; i.e., a code which minimizes redundancy will be an optimal perceptual code in the sense of efficiency when we consider *only* those basis functions which are maximally activated by a sensory stimulus. In reality, many different basis functions could be activated by the same sensory stimulus; if one function were consistently activated by a stimulus which activated some other function more strongly, these less-than-maximal activations would be wasteful. An optimal code must thus be designed not only to minimize redundancy in the Shannon sense; it must ensure that basis functions are activated either strongly or not at all. This requirement is embodied by another information theoretic measure called mutual information. The mutual information of two random variables X and Y, denoted $I(X;Y)$, measures the statistical dependence between those two variables. $I(X;Y)$ is symmetric, nonnegative, and equal to zero when X and Y are statistically independent. Mutual information is defined as shown in Equation 2.4, and, as shown in Equations 2.5 and 2.6 can be interpreted as either the difference between the individual entropies and the joint entropy, or as the reduction in the entropy of one variable given that the value of the second variable is known.

$$I(X;Y) = \sum_{x \in X, y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}, \tag{2.4}$$

$$= H_P(X) + H_P(Y) - H_P(X,Y), \tag{2.5}$$

$$= H_P(X) - H_P(Y|X). \tag{2.6}$$

2.5 Feature Selection

The principle of redundancy reduction states that a sensory coding system should seek to represent the sensory inputs in such a way that the sensory outputs are statistically independent from each other. The implications of this principle for feature selection are discussed

here. It is instructive to first consider feature selection methods used by the automatic speech recognition (ASR) community. ASR is a research field which has produced systems capable of converting audio inputs into useful outputs, and as a class of problems, bears more similarity to machine listening than a pure auditory analysis field like Computational Auditory Scene Analysis (CASA) or Blind Source Separation (BSS). We hence describe in detail a typical ASR feature selection scheme, relating it to the principles of redundancy reduction where appropriate, before considering feature selection for machine listening.

2.5.1 Feature Extraction for ASR

State-of-the-art ASR systems today typically use one of a few basic feature extraction techniques. The most popular is based on mel-frequency cepstral coefficients (MFCCs). The procedure for extracting MFCCs from audio involves the following steps:

Analog-to-Digital Conversion (ADC) ADC is the process of discretizing the continuous signal produced by the microphone. In ASR, ADC is typically performed at a sample rate of 16 kHz with 16-bit samples. This sample rate allows ASR systems to represent and analyze frequencies up to 8kHz, which is sufficient to capture the range of human speech. By contrast, some low-quality ADCs, such as telephone speech, have sample rates of 8 kHz with 8-bit samples, which is sufficient to capture *most* of the relevant range of human speech. Some cellular telephones use an 8 kHz sample rate with 16-bit samples.

Short-Time Fourier Transform (STFT) The Fourier transform is typically applied to sample windows of the ADC which have been passed through a Hamming window to avoid edge effects. A typical window size for ASR is 20 milliseconds, with 10 millisecond overlap. This window size is chosen as a tradeoff between frequency resolution (which is better with longer window sizes, particularly for energy in the lower frequencies) and temporal

resolution (which is better with shorter window sizes). The STFT process produces both a *power* spectrum and a *phase* spectrum for each sample window. The power spectrum represents the power of a signal at some frequency, while the phase spectrum represents relative timings within the sample window. In ASR and other audio applications, the phase spectrum is typically discarded. With a 16 kHz sample rate, the power spectra are 257 dimensional; the 20 ms / 10 ms window size results in a frame rate of 100 Hz.

Melscale Filterbank The melscale filterbank is a set of filters which was designed to crudely model the frequency responses of human perceptual responses. Each filter in a melscale filterbank is a bandpass filter whose width varies directly with the center frequency; that is, low-frequency filters have narrow passbands while high-frequency filters have wide passbands. In typical ASR systems, a 13-filter melscale filterbank is applied to power spectra from the STFT, reducing feature dimensionality by a factor of 20 while retaining important perceptual characteristics.

Mel-Frequency Cepstral Coefficients Cepstral coefficients are coefficients of the Fourier transform of the log power spectrum. They are used in ASR primarily because they lead to better performance than other features, and they are also robust across speakers, in part because they do not represent pitch information. In practice, MFCCs derived from mel spectra are typically used. The mel spectra are passed through a logarithmic nonlinearity, and then the discrete cosine transform (DCT) is applied. What is especially interesting about the DCT is that, for speech and speech-like signals, the DCT is a *decorrelating* transform. This means that the MFCCs are mutually decorrelated, or equivalently, that their covariance matrix Σ is diagonal. If the distributions over these features are Gaussian (i.e., the values of moments of third order and higher are zero), decorrelation is equivalent to statistical independence. As such, if the Gaussian assumption holds for a given dataset, MFCCs are optimal in a redundancy reduction sense, if not a recognition performance

sense.

Context In speech and other auditory phenomena, signals change rapidly with time, and they often do so in predictable and characteristic ways. As such, it is an attractive proposition to capture in each frame not just some theoretically optimal set of features, but also the *trajectory* of features from frame to frame. Adding context can take one of two forms: derivatives of the MFCCs can be explicitly computed, or windows of MFCC frames can be stacked together. In general, more context is better, and many ASR researchers stack 13 frames of MFCCs into a single feature vector. This stacking yields a feature space of 169 dimensions. This high dimensionality can lead to data sparseness problems which can be partially addressed with dimensionality reduction techniques.

Linear Discriminant Analysis Linear discriminant analysis (LDA) is one of a class of data transformation techniques which seek to project high-dimensional data onto some lower dimensional space in an optimal fashion. There are many definitions of "optimal" to consider; LDA seeks a transform in which data labeled as belonging to different classes are maximally linearly separable in the low-dimensional space. In the case of ASR, subphoneme level labels are available to guide the LDA process. Typical ASR systems use LDA transforms which bring the final feature vector to a few dozen features.

Normalization The above steps represent the most *basic* ASR feature extraction process. Typically, there are many normalization steps involved which aim to remove the differences between speakers, microphones, and environments as much as possible.

2.5.2 Feature Extraction for Machine Listening

In machine listening, the optimal feature set is not clear. Some applications have successfully employed MFCCs, while others have made use of more esoteric features. It is

likely that the optimal feature set in terms of system performance is strongly application-dependent. It is worthwhile to first explore the use of traditional ASR features for machine listening tasks for a number of reasons. Melscale spectra are worth considering mainly because they approximate human frequency response; in this sense they theoretically contain all the information needed to perform all the listening tasks that humans can perform. MFCCs are worth considering because they represent a decorrelating transform; under a Gaussian assumption they are thus optimal in a redundancy reduction sense. In addition, they are very useful for speech recognition. As Lewicki pointed out in [58], speech is made up of both harmonic and non-harmonic sounds (i.e., vowels and consonants) and exploits the entire range of human hearing. Non-speech environmental sounds are also made up of harmonic and non-harmonic sounds; as such, MFCCs should also be a good general representation of arbitrary environmental sounds. They may not be optimal in a performance sense for some machine listening tasks, however, mainly because they do not preserve pitch information which may be important in some listening tasks like general sound recognition.

In addition to melscale spectra and MFCCs, we consider a number of additional features in this thesis. These features have sometimes been referred to as "perceptual" features, though they do not necessarily correspond to features humans use in perceptual tasks. Many of them are simply convenient summaries of gross spectral characteristics, some of which should intuitively be useful for certain listening tasks. These features are described below. In what follows, $p_i t$ refers to the i^{th} of M power spectral coefficients in frame t, $a_j t$ refers to the j^{th} of N sample points in the t^{th} sample window in the time signal, and ϑ is a generic threshold value.

Loudness Changes in perceptual loudness are often indicative of changes in the auditory scene; e.g., event onsets or offsets and changes of environment. Perceptual loudness can be grossly approximated by measuring signal energy. One can measure energy in either the time domain or the frequency domain. A time-domain energy measure, power, is given in

Equation 2.7, while a frequency-domain energy measure, root mean square energy, is given in Equation 2.8.

$$POW(t) = \sqrt{\frac{1}{N}\sum_{j=1}^{N} a_j(t)^2}. \tag{2.7}$$

$$RMSE(t) = \sqrt{\frac{1}{M}\sum_{i=1}^{M} p_i(t)^2}. \tag{2.8}$$

Bandwidth Bandwidth is a measure of the extent of a sound in the frequency domain. Tonal sounds often have relatively narrow bandwidth, while non-tonal sounds like transients often have relatively wide bandwidth. Sound fields can also have characteristic bandwidths. Bandwidth, measured in Hz, is calculated as the difference between the highest frequency exceeding some energy and the lowest frequency exceeding the same threshold, as shown in Equation 2.9.

$$BW(t, \vartheta) = \arg\max_{i=1}^{M}(p_i(t) > \vartheta) - \arg\min_{i=1}^{N}(p_i(t) > \vartheta). \tag{2.9}$$

Spectral Diffusion Spectral diffusion is, like bandwidth, a measure of the extent of a sound in the frequency domain. However, whereas bandwidth is a simple difference between two frequencies, diffusion measures the actual spread of energy among all frequencies. This difference can be illustrated by considering a sound field consisting of two pure tones, one at a low frequency and one at a high frequency. This sound field will have a high bandwidth value, but low spectral diffusion. Diffusion is similar to a discrete entropy measure of the energy over frequency space and is shown in Equation 2.10.

$$SD(t) = \sum_{i=1}^{M} \frac{p_i(t)}{\sum_{j=1}^{M} p_j(t)} \log \frac{p_i(t)}{\sum_{j=1}^{M} p_j(t)}. \tag{2.10}$$

Spectral Centroid Spectral centroid is a measure of spectral location; it is the weighted mean frequency of the spectral energy, measured in Hz. Centroid is related to the zero-crossing rate (ZCR) measure which is used for speech detection in some ASR systems. Centroid also corresponds loosely with the perceptual experience of brightness; sounds with a high spectral centroid tend to stand out in the sound field. Further, noisy or chaotic sound fields (e.g., traffic) tend to dampen the overall spectral centroid; hence it can be a useful measure for both sound recognition and for environment recognition. The spectral centroid is shown in Equation 2.11.

$$SC(t) = \frac{\sum_{i=1}^{M} ip_i(t)}{\sum_{i=1}^{M} p_i(t)}. \tag{2.11}$$

Band Energy Ratio Band energy ratio measures the relative spectral energy between frequency bands up to some threshold frequency and frequency bands above the threshold. This feature has been used to distinguish between voiced sounds (i.e., vowel portions of speech) and non-voiced sounds and is thus ideal for identifying speech and sound fields made up primarily of babble noise. A threshold at 1.5 kHz is reportedly optimal for this task; other thresholds could potentially be useful for distinguishing between other broad sound classes. Band energy ratio is shown in Equation 2.12.

$$BER(t, J) = \frac{\sum_{i=1}^{J-1} p_i(t)^2}{\sum_{i=J}^{M} p_i(t)^2}. \tag{2.12}$$

Signal-to-Noise Ratio Signal-to-noise ratio measures the relative strength of noisy and non-noisy sound components in a given analysis frame. SNR is shown in Equation 2.13. Here, the STFT is computed and estimates of the noisy components vs. signal-containing components of the power spectra are made using a technique from Ephraim and Malah [31], also used by Westphal [110], [111] to estimate and remove noise in real-time for ASR in automotive environments.

$$SNR(t) = 10 \log \frac{POW_{signal}(t)}{POW_{noise}(t)}. \tag{2.13}$$

Component Analysis

We also consider context in some applications; as in ASR, this choice often results in feature spaces which are inconveniently large. We thus need to explore dimensionality-reducing feature space transformations like LDA. However, for general listening tasks, we often lack detailed labels of the type ASR researchers can use to produce good LDA transformations. We thus focus on techniques which seek to optimally *represent* the high-dimensional data in the sense of reconstructing the original data with minimal error. The two transformations we use in this thesis are principal component analysis (PCA) and independent component analysis (ICA). Both PCA and ICA assume a model in which some underlying signal $\mathbf{s}$ is modified by some filter $\mathbf{A}$, resulting in an observable signal $\mathbf{x}$ as shown in Equation 2.14.

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{2.14}$$

The filter $\mathbf{A}$ is usually referred to as the *mixing matrix*. The task of PCA and ICA is to find an estimate $\mathbf{W}$ of the *unmixing matrix* $\mathbf{A}^{-1}$ such that we can recover an estimate $\hat{\mathbf{s}}$ of the original signal $\mathbf{s}$ as shown in Equation 2.15.

$$\hat{\mathbf{s}} = \mathbf{x}\mathbf{W}. \tag{2.15}$$

Theoretically, $\hat{\mathbf{s}}$ is an estimate of the underlying sources in the sound field prior to the effects of mixing. The rows of the unmixing matrix $\mathbf{W}$ are referred to as the *basis functions* of the transform; the values of $\hat{\mathbf{s}}$ are taken to be coefficients of the relative strength of each basis function.

PCA and ICA define optimality in different ways. These definitions and their consequences are discussed below.

Principal Component Analysis Principal Component Analysis seeks an unmixing matrix $\mathbf{W}$ for which the following three properties hold. First, the covariance matrix of the new feature set $\hat{\mathbf{s}}$ should be diagonal; that is, the features $\hat{s}_i$ should be mutually decorrelated. Second, the basis functions should be ordered by variance. Third, each basis function should be orthogonal to neighboring basis functions. A transform of this type can be calculated by computing the eigenvectors of the covariance matrix of the observed feature space, Σ_x; each eigenvector is a basis function. The eigenvalues of this system correspond to variance in the new feature space $\hat{\mathbf{s}}$ and can be used to order the basis functions, and also to reduce the dimensionality of $\hat{\mathbf{s}}$; i.e., by omitting basis functions with small eigenvalues. $\mathbf{W}$ can also be estimated using the singular value decomposition (SVD) directly on the (non-square) observed features $\mathbf{x}$. In SVD, a matrix $\mathbf{M}$ is factored as $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^*$; the matrix $\mathbf{U}$ contains the orthonormal basis functions of $\mathbf{M}$, while Σ contains the singular values, which can be used in a manner similar to the eigenvalues of the eigenvalue decomposition. A final, non-analytical method for estimating $\mathbf{W}$ is to train an autoencoding multilayer perceptron (MLP). When such a neural network is trained with the input data as the target and a mean-squared error metric, the hidden layer weights will converge to a matrix spanning the same subspace as the analytical PCA solution, up to a rotation.

Using any of the above approaches, PCA will learn a *decorrelating* transform. If the underlying data are Gaussian, decorrelation is equivalent to independence; hence PCA can be viewed as a redundancy reduction transform under the right conditions.

Independent Component Analysis Independent Component Analysis seeks an unmixing matrix $\mathbf{W}$ for which the features in the transformed feature space $\hat{s}_i$ are mutually independent; i.e., $I(\hat{\mathbf{s}}) = 0$. No analytical algorithm exists to recover an unmixing matrix which

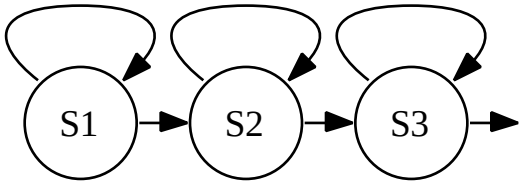

Figure 2.1: A Hidden Markov Model (HMM)

fulfills this requirement; all ICA approaches rely on iterative optimization techniques and none are guaranteed to find a globally optimal solution. As noted above, many different criteria and algorithms have been suggested for ICA; Hyvärinen and Oja's FastICA package [48] implements their popular approach, which is used in this thesis.

2.6 Time Series Modeling for Audio

Given some feature set, a common task in this thesis, and in audio processing in general, is how to convert the audio signal into useful symbolic information. Many pattern recognition techniques exist which can make this conversion. The dominant model for ASR and audio in general, however, is the hidden Markov model (HMM) [81]. The HMM is a finite-state machine in which each hidden state emits observable values, either discrete symbols (which can be modeled by simple discrete probability distributions) or real-valued vectors (which are typically modeled by mixtures of Gaussian distributions). Each state corresponds to some symbolic value. For example, in ASR, each state might represent a phoneme. Then, given a set of audio feature vectors, using HMMs, one is able to *infer* the sequence of phonemes which is most likely to have produced that sequence of vectors. An 3-state HMM with a forward topology, the kind most frequently used in ASR, is shown in Figure 2.1.

It can also be instructive to view HMMs in terms of dependencies among random variables; under this viewpoint an HMM is just a specific kind of dynamic Bayesian network

$$S_t \leftarrow S_{t-1}, \tag{2.16}$$

$$O_t \leftarrow S_t. \tag{2.17}$$

Figure 2.2: HMM Dependencies

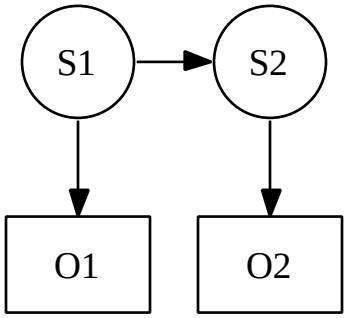

Figure 2.3: DBN Representation of HMM Dependencies

(DBN). An HMM has two variables: the state S and the output O. In HMMs, the state at time t, S_t, is dependent *only* on the state at time $t-1$, and the output at time t, O_t, is dependent only on the state at time t. These dependencies are shown more formally in Figure 2.2 and graphically in Figure 2.3. For more detailed information on the benefits of viewing HMMs as DBNs, as well as an excellent overview of DBNs in general, the reader is referred to Murphy's Ph.D. thesis. [71].

In order to apply HMMs to some pattern recognition task, there are three problems to consider:

1. Given an observation sequence $\mathbf{O}$ and an HMM $\mathbf{M}$, how can the probability of the observation given the model, $P(\mathbf{O}|\mathbf{M})$, be estimated?

2. Given an observation sequence $\mathbf{O}$ and an HMM $\mathbf{M}$, how can the state sequence $\mathbf{q}$ *most likely* to have generated $\mathbf{O}$ be estimated?

3. How can the parameters of the HMM $\mathbf{M}$ be adjusted to maximize $P(\mathbf{O}|\mathbf{M})$?

The first two problems, probability estimation and inference, are solved by dynamic programming procedures: the forward procedure and the Viterbi algorithm. The third problem is solved by the Baum-Welch procedure, an expectation maximization (EM) procedure in which expected counts for all states and all outputs at all time steps are accumulated using the forward and backward procedures and subsequently converted to probabilities. An alternate optimization procedure in common use involves substituting forward and backward expectations with Viterbi expectations. The reader is referred to the classic HMM tutorial by Rabiner [80], as well as ASR texts by Rabiner and Juang [81] and Jelinek [52] for more information on these procedures.

There is a fourth problem for HMMs which is not typically considered by most ASR researchers, but which is vital for general audio modeling. This problem is how to infer the structure of the HMM from data, in the absence of detailed prior knowledge. Structure learning is often not a problem in ASR and similar applications, because domain knowledge allows researchers to make informed decisions about the appropriate topologies to use. In general audio modeling, it is often not clear what kinds of topologies are appropriate to model the phenomena of interest. Hence, one must seek methods for learning topologies, as well as parameters, from unsegmented data. Approaches of note toward this goal include Stolcke and Omohundro's model merging method [97], Brand's parameter extinction method [9], Frietag and McCallum's stochastic optimization method, and Reyes-Gomez and Ellis' leader-follower clustering method [83].

2.7 Chapter Summary

I presented in this chapter an overview of the theoretical background of machine listening, covering the nature of the audio signal, basic audio analysis and feature extraction, and the

organizing principles of general machine perception. I concluded with an introduction to the main tool for modeling time series in audio, the hidden Markov model. The material presented here constitutes sufficient background to understand the remainder of this thesis.

Chapter 3

Auditory Gisting for Summarization and Stream-of-Interest Selection in Multimedia

3.1 Chapter Overview

This chapter describes theoretical and practical work on multimedia summarization and stream-of-interest selection using audio data. Specifically, it demonstrates how the relative value of segments of multimedia can be estimated using only information theoretic measures of the information content of the audio streams. I show that this approach, called blind value assignment (BVA), can be used to produce value judgments that are broadly similar to human value judgments in two specific tasks — after-action review generation generation (AARG), a summarization task, and online aggregate remote sensing (OARS), a stream-of-interest selection task. I begin the chapter with a discussion of the motivation for this work, and why BVA is so well-suited for applications of this kind. After discussing prior work on the closely-related field of blind segmentation, I present the BVA metrics

used in this thesis, followed by experiments on the AARG and OARS problems which demonstrate the effectiveness of the approach.

3.2 Multimedia Summarization

As noted in Chapter 1, there has recently been a rapid proliferation of inexpensive devices, both stationary and portable, which are able to collect and store massive amounts of multimedia sensory data. As a result, the ability to construct rich, detailed accounts of human activity has also grown. This capability continues to grow and the pace of growth will only accelerate as time passes. One can easily imagine a state in which these massive corpora can be used to collect intelligence, augment human memory, tell stories, and share experiences. One can equally easily imagine a state in which these corpora are rendered unwieldy by their size and sparseness with respect to interesting and useful events. Much of daily human life is spent on repetitive, boring tasks that are not worth reviewing or reporting; truly interesting and useful events are relatively rare. Humans are largely incapable of finding them in large databases, and machines which need to process large collections of multimedia corpora could also benefit from some form of preprocessing so they can focus their resources on areas of genuine interest. For this reason, systems capable of providing a low-resolution means for humans and machines to conduct exploratory data analysis is needed. In the case of multimedia data, this summarization process is often referred to as skimming or gisting.

A multimedia summarization system produces, from some multimedia corpus, a selection of segments that in some way captures the interesting aspects of the corpus. In this way, a human or machine user can rapidly peruse the highlights of the corpus and achieve a high-level understanding of the events, environments, and activities that it contains. Further, the summary can itself serve as a platform for deeper analysis. Upon finding a particularly interesting scene, the user can query the system to discover other, similar

scenes. In practice, a summary to be used in this manner should consist of an ordering over all segments, such that the user can select an absolute or relative summary size.

This kind of summarization is distinct from query-specific summarization in that it does not rank segments according to their utility compared to some user-supplied specification. It is instead completely undirected and thus not dependent on specific event or activity models. This mode of operation is complementary to model-based analysis and relies on the statistics of the signal to estimate value.

3.3 Blind Segmentation and Value Assignment

The process of creating the kind of general summary of interest in this work is often referred to as a value assignment problem. That is, the process of summarization can be viewed as the process of finding those segments in a multimedia stream with high value, for some definition of value. In some applications, the definition of value is quite specific, and certain classes of events are known to be of interest *a priori*. In this case, if enough data are available, relevant models can be trained and value assignment becomes a detection problem. Examples of this situation include broadcast news speech recognition (in which speech regions have value, while music regions do not) and office activity recognition (in which specific sounds or events can be important context cues).

In the applications envisioned here, however, there is generally no such prior expectation of the kinds of events which might be of interest, and hence no models can be built to recognize and detect these events. For this reason, these kinds of applications are referred to as *blind* segmentation or value assignment problems. While blindness means that there is no way to bias value assignments toward specific classes of events or scenes, it does provide robustness to noisy or unseen conditions under which semantic models may break down. Since the statistics of the signal are all that a blind method has to work with, these concerns simply do not matter. Furthermore, methods based solely on the statistics of the

signal rather than potentially complex semantic event models are fast, making them well-suited to real-time operation, exploratory data analysis, and deployment on devices with limited computational power.

In this work, audio data is the focus of value assignment, even in cases where video data is available. This is a reasonable choice for many of the reasons given in Chapter 1. To reiterate these reasons, the audio signal has low bandwidth and processing costs, it is omnidirectional, and it is immune to occlusion, sensor motion, and changes in lighting conditions. Finally, events which are of import in the real world often leave behind clear acoustic evidence which is easy to detect even with blind methods.

3.4 Prior Work

There is a large body of work on video-based segmentation and summarization of multimedia data, much of it focused on rather esoteric, application-specific features like scene breaks in film and televised news and sporting events. See the survey by Lienhart ([60]) for general information on this topic. Gaborski et. al. recently demonstrated results on blind novelty detection in video; see [105] and [37]). Also of interest is the work of Itti and colleagues. Their work ([51], [50], [86]) demonstrated, via tracking of human saccade behavior when exposed to video, that humans prefer to pay attention to those regions where local entropy and local surprise are high. By estimating and tracking these regions of high interest using information theoretic constructions like saliency and relative entropy, Itti et. al. can construct a visual gist of a scene. Itti's notion of gist shares some features with the blind value assignment approach developed below; specifically, the metrics are quite similar. However, the focus on visual regions rather than temporal regions shows a quite different application goal. Also related to the goals of this thesis is work from Oliva, Torralba, and collaborators. Their focus on low-level, global statistics of images in [102], [103], [76], and [104] allows for extremely efficient capture of relevant characteristics of

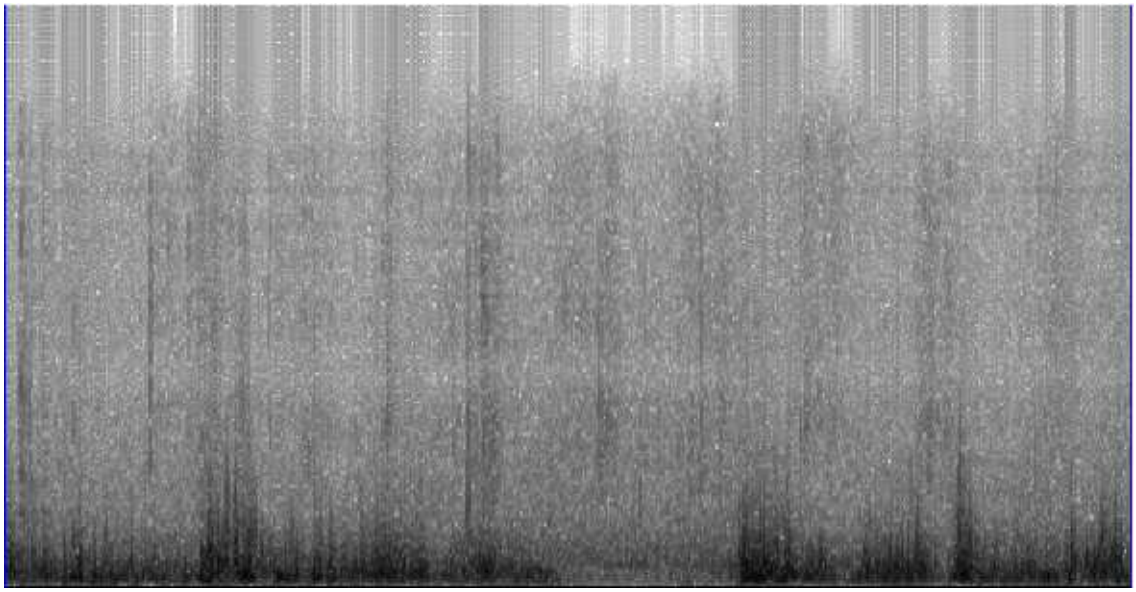

Figure 3.1: Log Power Spectra Example

an image, or a sequence of images. In addition to applications like environment or locale recognition, these methods could also in principle apply to the gisting domain.

In the audio domain, Foote proposed in [34] and [35] a method for blind segmentation based on self-similarity, and also showed, with Cooper et. al. in [23] that this method could be used for shot boundary detection in video. In this approach, for a given T-length sequence of feature vectors $\mathbf{S}$, a $T \times T$ self-similarity matrix A is created in which each element A_{ij} is the similarity between frames S_i and S_j. Foote used the cosine measure, $\frac{S_i \cdot S_j}{||S_i|| ||S_j||}$, to measure similarity. The resulting matrix has many interesting properties and yields an easily visible representation of rhythm and repeated substructures, making it especially well-suited to music representation. By way of example, Figure 3.1 shows the power spectra (in log space for ease of viewing) for 5 seconds of environmental audio; the corresponding self-similarity matrix is shown in Figure 3.2.

The main use of the self-similarity matrix is for scene change detection. To detect scene

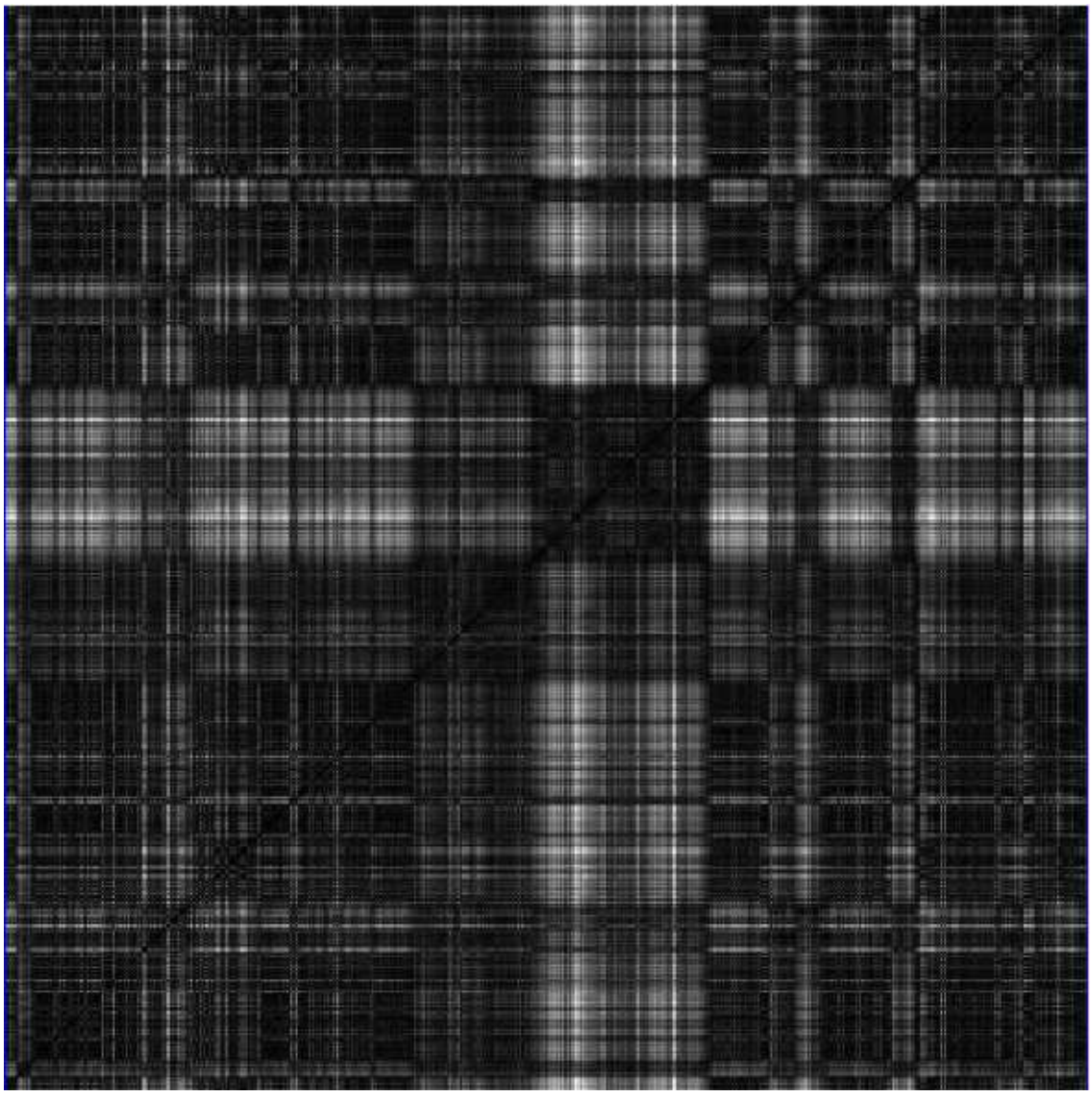

Figure 3.2: Temporal Structure of Log Power Spectra, Example per Foote

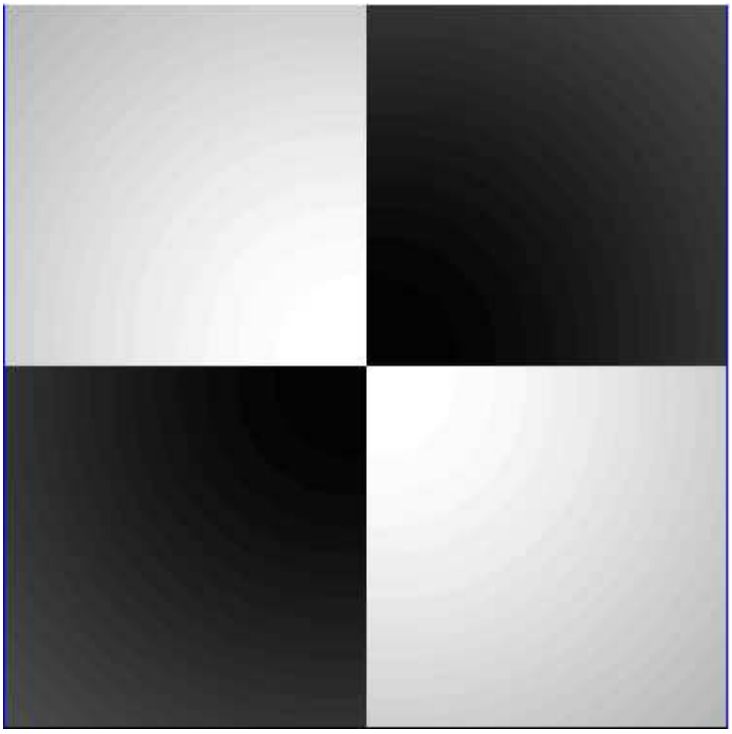

Figure 3.3: Scene Change Kernel per Foote

changes in this framework, Foot suggested the use of a scene-change kernel K, a matrix encoding a perfect scene change. This matrix has four square subregions whose values are given as:

$$K_{ij}(N) = \begin{cases} 1 & \text{if } (i < \frac{N}{2}) = (j < \frac{N}{2}) \\ -1 & \text{otherwise} \end{cases} \tag{3.1}$$

In practice, this matrix is smoothed by a Gaussian function to focus on the center of the kernel; an example of this kernel is shown in Figure 3.3. Also, since this matrix is symmetric, it can be represented more efficiently in what Foote calls the "slant domain," which considers only the values above the diagonal. A slant-domain scene-change kernel is shown in Figure 3.4.

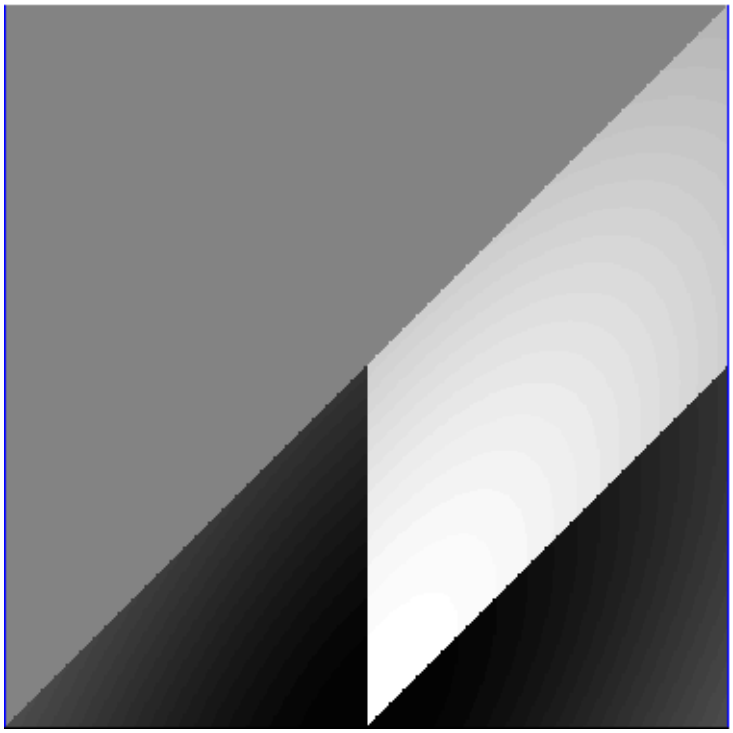

Figure 3.4: Scene Change Kernel per Foote, Slant Domain

The scene change kernel can be used to detect scene changes by multiplying it with a given segment's self-similarity matrix; the resulting novelty measure for segment $\mathbf{S}$ is given as:

$$Nov(S) = \sum_{ij} A_{ij} K_{ij}, \tag{3.2}$$

where A is the self-similarity matrix of $\mathbf{S}$. Foote suggested that hierarchical segmentations could be produced by computing novelties with varying kernel sizes. He also suggested that auditory summaries could be produced by first, given some novelty threshold, segmenting an audio recording, and then, for each segment, selecting the subsegment that is most similar to the segment as a whole. This approach to summarization was proposed in the musical domain, where it is important to summarize a piece with one of its more recognizable structures. It is not clear that this approach is useful when dealing with less-structured, more event-sparse data — a given segment in unstructured real-world audio is likely to be dominated by silence or unidentifiable noise.

Slaney, in [92] and [91], presented the scale-space delta approach to multimedia segmentation. This work, like Foote's, presented a hierarchical blind segmentation method. This method first involved computing smoothed, or scale-space versions, of the input features $\mathbf{S}$. These smoothed features, $\bar{\mathbf{S}}$, are computed as:

$$\bar{S}_t^{\sigma} = \int_{\tau=-\infty}^{\infty} S_{\tau} e^{\frac{-\tau^2}{2\sigma^2}}, \tag{3.3}$$

where σ is a smoothing factor. In Slaney's approach, N different values of σ are used, and for each σ, the first derivatives $\Delta\bar{\mathbf{S}}^{\sigma}$ are computed and summed. The result is the scale-space delta feature $\Delta\bar{\mathbf{S}}$. Using the same example power spectra above, the scale-space delta feature is shown in Figure 3.5.

After the scale-space delta feature has been computed, peaks are found for each of the N values of σ, as shown in Figure 3.6. Peaks for small values of σ correspond to

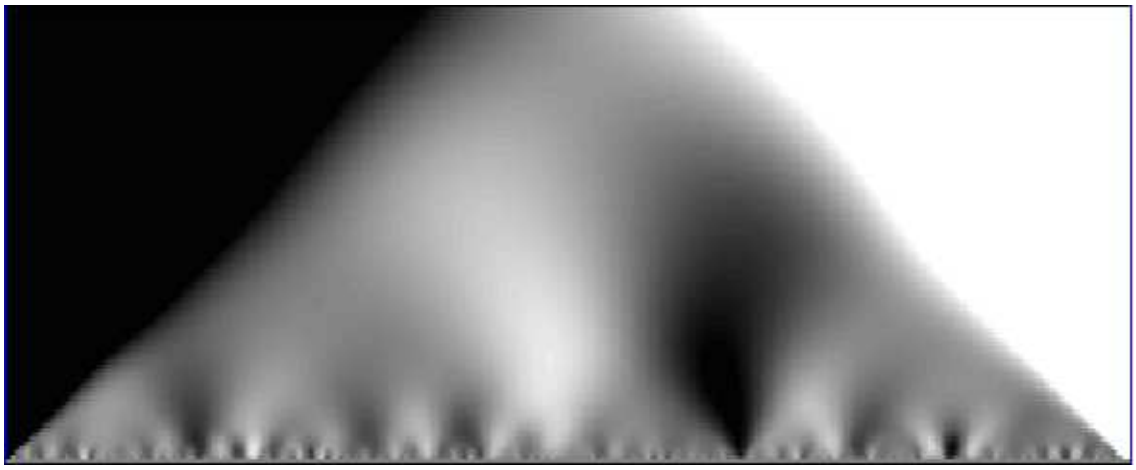

Figure 3.5: Scale-Space Delta Sums of Log Power Spectra, σ=0-100, per Slaney

short-term scene changes, while peaks for large values of σ correspond to long-term scene changes. By using dynamic programming techniques, long-term peaks can be tracked back to indices in the deltas of the unsmoothed feature, and hence to real time indices. Slaney showed that this method was effective for segmentation of multimedia news broadcasts and documentaries.

Chen et. al. used the Bayesian information criterion (BIC) to propose in [14] the Δ-BIC method for blind segmentation. The BIC, computed as

$$BIC(X) = -2 \ln L(X) + k \ln |X|, \tag{3.4}$$

where $L(X)$ is the likelihood of the data given some model, k is the number of model parameters, and $|X|$ is the number of observations, is a measure of model quality which rewards good data fit but penalizes many parameters. In this Δ-BIC approach, for a proposed segmentation point in a data set, the BIC of representing the data with a single model is compared to the BIC of representing it with two models, one to either side of the segmentation point.

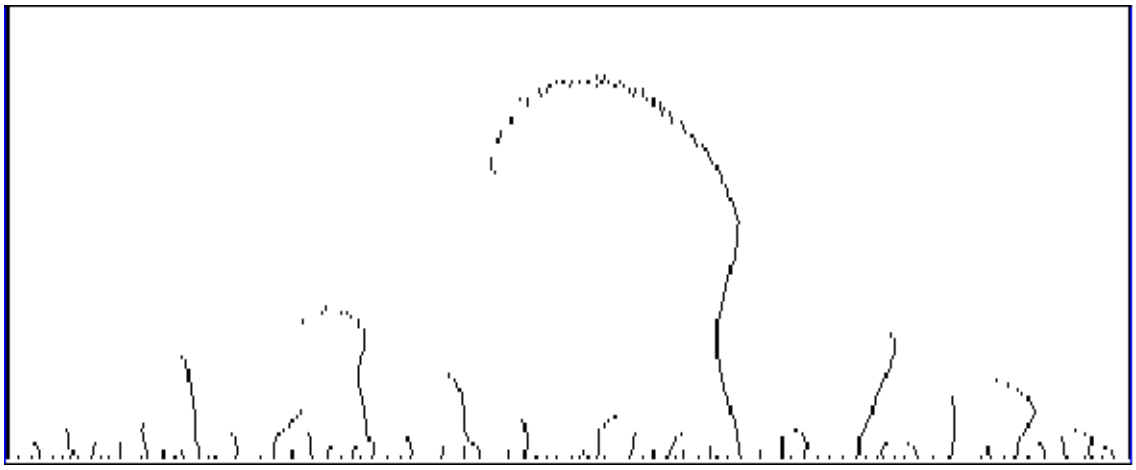

Figure 3.6: Scale-Space Delta Sum Peaks, σ=0-100, per Slaney

Finally, Siegler et. al. proposed in [87] the KL2, or symmetrized relative entropy metric, to segment broadcast news data for automatic speech recognition. This method was shown effective for this segmentation task, and is in fact the basis for the blind value assignment method discussed below.

3.5 A Metric for Blind Value Assignment

As in the works cited above, we consider information-theoretic metrics for BVA. In that these kinds of metrics can be of use in finding events which are rare or highly informative in some technical sense, it is possible in principle to use them as proxies for the more intuitive senses of interesting or informative. The information content of a random variable in isolation can be estimated be calculating its entropy (see Chapter 2). However, in multimedia summarization applications, the goal is to produce *relative* rankings of value, and to use these rankings to produce concise summaries of content. If segments were simply to be ranked by entropy, it is possible that many very similar segments might be ranked highly and included in the summary. This kind of redundancy is to be avoided both on

practical and theoretical grounds. Practically speaking, in most cases, humans interested in exploring a large dataset are more interested in experiencing the full breadth of scenes and events rather than a sequence of identical or similar scenes. Theoretically speaking, a summary with many similar segments is redundant and so wastes the capacity of the summary channel. It is thus important to view multimedia summarization as a redundancy reduction problem.

Given this view, the proper way to assign value is in such a way that the entropy of the *summary* is maximized, or, equivalently, that the segments included in the summary are maximally mutually statistically independent. As there are $\left(\begin{array}{c} n \\ m \end{array} \right) = \frac{n!}{n-m!}$ possible m-element summaries of an n-segment stream, it is impractical to search the entire space for the one which maximizes entropy. Instead, a greedy approach can be implemented in which, k segments already ranked in the summary, the $k + 1^{th}$ segment is the segment that is maximally surprising. One measure of surprise, or equivalently, of model quality, is the cross entropy, given in Equation 3.5.

$$H_{p,q}(X) = -\sum_{x \in X} p(x) \log q(x). \tag{3.5}$$

Cross entropy can be interpreted as the cost of encoding events from one distribution, q, with the *incorrect* distribution p. If p is the model used to generate the k segments already ranked in the summary, and q is the model used to generate segment $k + 1$, then cross entropy can be used as a ranking metric. One issue with this metric, however, is that the inherent difficulty of coding segment $k + 1$ is not considered. That is, it is possible that $H_{p,q}(X_{k+1})$ is high largely because X_{k+1} itself has high entropy. In order to account for this fact, we instead normalize the cross entropy by subtracting the entropy of the candidate segment; the resulting measure is called the relative entropy, or Kullback-Liebler divergence, and is given in Equation 3.6. Relative entropy can be interpreted as the coding effort that is *wasted* by assuming that the data X was generated by distribution p instead of

the correct distribution q [24].

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}, \tag{3.6}$$

$$= H_{p,q}(X) - H_p(X). \tag{3.7}$$

A further problem with relative entropy is that it is not symmetric; this difficulty can be repaired by considering the symmetrized version given in Equation 3.8. This measure, which first appeared in [87], is just the sum of the relative entropies $D(p||q)$ and $D(q||p)$. One added normalization which can be made is to divide the relative entropies by the cross entropy as shown in Equation 3.10; the resulting metric, shown in Equation 3.11, is always between 0 and 1 and is interpreted as the average percentage of coding cost wasted by assuming an incorrect distribution. This metric, $\hat{D}^2$, or normalized, symmetrized relative entropy (NSRE), is used as the main metric in all following value assignment exercises .

$$D^2(p||q) = D(p||q) + D(q||p), \tag{3.8}$$

$$= H_{p,q}(X) + H_{q,p}(X) - H_p(X) - H_q(X). \tag{3.9}$$

$$\hat{D}^2(p||q) = \frac{D(p||q)}{H_{p,q}(X)} + \frac{D(q||p)}{H_{q,p}(X)}, \tag{3.10}$$

$$= 2 - \frac{H_p(X)}{H_{p,q}(X)} - \frac{H_q(X)}{H_{q,p}(X)}. \tag{3.11}$$

3.6 Evaluating The Auditory BVA Approach

To evaluate the auditory BVA approach for multimedia summarization and stream-of-interest selection, experiments were designed in which real audiovisual data were collected

and human judgments for both tasks were elicited. These judgments were used as a baseline against which to evaluate automatic value judgments based on the metrics described above and the specific application-dependent procedures discussed below.

3.6.1 Data Collection

Data for this evaluation were collected using a Hitachi MPEG-1 video camera attached to the author's backpack while he carried out a number of errands on and around the CMU campus. These tasks included:

Stream 1 Visiting an ATM to make a withdrawal. See Figure 3.7.

Stream 2 Mailing a letter and purchasing a soda from a vending machine. See Figure 3.8 and Figure 3.9.

Stream 3 Buying lunch from a mobile vendor. See Figure 3.10.

Stream 4 Taking a car to a gas station for a fillup. See Figure 3.11.

The streams ranged in length from 10 to 15 minutes. An external microphone was not used for this study, as the onboard microphone was judged to be of sufficient quality to capture the relevant sounds. Audio was extracted from the MPEG-1 video with a sample rate of 16 kHz and a sample depth of 2 bytes.

3.6.2 Feature Extraction and Data Modeling

Three separate audio feature sets were used in this evaluation, along with two sets of video features for comparison. All audio features were computed at a frame rate of 100 per second. The audio features used included a 16-dimensional set of melscale spectra (MEL), a set of 16 MFCCs derived from those spectra, and a set of four spectral summary features (SPEC) including spectral centroid (see Equation 2.11), root mean square energy (see

Figure 3.7: Visiting an ATM

Figure 3.8: Mailing a Letter

Figure 3.9: Purchasing a Soda

Figure 3.10: Purchasing Lunch from a Mobile Vendor

Figure 3.11: Filling the Car with Gasoline

Equation 2.8), signal-to-noise ratio (see Equation 2.13), and a band energy ratio at 200 Hz (see Equation 2.12). In these experiments, the two 20-dimensional feature sets MEL+SPEC and MFCC+SPEC were evaluated. The first video feature used was a set of color features (COLOR) made up of 128 histogram values in the hue, saturation, value (HSV) color space, sampled at a rate of 30 frames per second. The second video feature used was a set of motion features (MOTION). The motion features consisted of five values: energy, horizontal motion mean, vertical motion mean, horizontal motion variance, and vertical motion variance. These features were intended to characterize both global camera motion and local object motion. These features were computed based on optical flow outliers every three video frames using the Lucas-Kanade algorithm [61], resulting in a frame rate of 10 per second.

After feature extraction, the data in this study was represented as a set of real-valued vectors, and not sequences of discrete values as implied by all the metrics discussed so far. Real-valued versions of information theoretic measures (often termed *differential* entropy, etc.) do exist [1]; however, unlike in the discrete case, the distribution P must be known in order to compute these measures correctly. In practice, this work uses the simplifying assumption that all random variables were generated by multidimensional Gaussian distributions with diagonal covariance matrices. This assumption is clearly false; however, it can be shown that for any given covariance matrix K, the Gaussian distribution is the maximal-entropy distribution with that covariance matrix [24]. In other words, employing this assumption amounts to calculating upper bounds on the various metrics given the data.

In order to compute the NSRE for data modeled in this way, it is necessary only to be able to compute the entropy of a Gaussian distribution with covariance matrix $|K|$ and the cross entropy between two Gaussian distributions with covariance matrices $|K_Q|$ and $|K_P|$. The Gaussian differential entropy is given in Equation 3.12; the Gaussian differential cross

[1] These real versions generally have strange properties; the differential entropy $h_P(X)$, for instance, can be negative!

entropy in Equation 3.13.

$$h_P(\mathbf{X}) = \frac{1}{2}\ln(2\pi e)^n|K|. \tag{3.12}$$

$$h_{P,Q}(\mathbf{X}) = \frac{1}{2}[\ln(2\pi e)^n|K_Q| + K_P K_Q^{-1} + (\mu_P - \mu_Q)^T K_Q^{-1}(\mu_P - \mu_Q)]. \tag{3.13}$$

3.6.3 The Evaluation Procedure

For each application, a simple value judgment interface was designed which allowed human subjects to either rank all the segments in each stream (for summarization) or to select, for each possible segment index, which of the four streams was most interesting at that time. The subjects were told that their goal was in the realm of coordinating a group activity in which each group member had a specific task which was part of the overall group aim but were not explicitly told about the contents of each stream. Different instructions applied to each task. Subjects in the summarization experiment were told that they were to create a summary of each stream such that other human users could view them and rapidly attain familiarity with their contents. Subjects in the stream-of-interest selection experiment were told that they were to monitor the events in real time and to select the one stream at each time index that most impacted their understanding of the entire operation.

Agreement with human judgments was used as the evaluation metric in this work. This metric was chosen in lieu of eliciting direct human evaluations of BVA judgments for two reasons. First, a notion like "quality of summary" is extremely hard for humans to describe quantitatively or measure. Second, humans may have pre-conceived notions about the ability of computational systems to pick out important aspects of sensory data; biases in either direction in this way could make the responses hard to interpret and could possibly require double-blind studies in which some subjects were given human-produced gists and others computer-produced gists. Simply asking humans to produce their own value assignments

sidesteps most of these problems. Under this regime, match rate is an easy-to-interpret metric. Further, since humans are not asked to grade the performance of some other entity, the ability to consciously or unconsciously exhibit bias in their responses never presents itself. One issue that does present itself is that humans themselves might not agree in their value judgments. One might expect a certain amount of disagreement, but one might also expect that certain segments would *always* be rated highly valuable by humans, simply because it was patently obvious that they contained useful information. The segment where the author actually gets cash from the ATM, for example, should be chosen by everyone. Finally, the ability to measure inter-human agreement allows one to specifically consider machine performance on those segments about which humans *did* agree, which one could argue is more important than matching human judgments in the absence of broad agreement.

3.7 Application: Multimedia Summarization for After-Action Review Generation

The goal of a summarization application is simply to present a shortened version of some multimedia stream which captures the information needed to attain an understanding of the important actions in the stream, given some limited channel capacity. In this thesis, such applications are referred to as after-action review generation (AARG) applications.

Given a multimedia stream, the target summary is an ordering of segments from most valuable to least valuable, assuming the information-theoretic definitions of value given above. In practice, the summary is kept in temporal order for human perusal. To construct this ordering, a greedy algorithm is used. The summary is first seeded with the segment which has the highest entropy; subsequent segments are added to the summary in order of descending NSRE between candidate segments and segments already in the summary.

In addition to this greedy algorithm, an alternate approach based solely on scene changes

Stream	Segments
1	71
2	69
3	90
4	89
Average	79.8

Table 3.1: Number of 10-second Segments per Stream

was tested. In this approach, the NSRE was used *internally* on each segment to estimate to what degree that segment contained a scene change; this approach is similar to the blind segmentation work discussed in Section 3.4. In this experiment, five scene change measurements were taken per segment and the maximum was used as the scene change value. In order to create a summary using this technique, the segments were ranked by strength of scene change alone, with no consideration given to similarity to other segments.

Experiments were carried out by first eliciting human value judgments and then comparing BVA judgments against this standard. In order to simplify both the value elicitation process and the evaluation process, all streams were split into 10-second segments; the number of segments per stream is shown in Table 3.1. In this way, value elicitation proceeded at a faster pace, and deciding whether a machine judgment matched a human judgment became a matter of checking whether or not segment indices matched. In order to actually elicit these judgments, the streams were converted to RealMedia format using the RealProducer and RealMedia Editor packages [82] in conjunction with the SMIL multimedia markup language [94]. A tool, shown in Figure 3.12, was constructed which allowed humans to play any segment at any time, and to populate a summary using these segments. The summary was also playable at any time, allowing humans to add and remove segments and evaluate the effects in real time.

Figure 3.12: The AARG Gist Construction Tool

Subjects were asked to view each stream in its entirety at least once, and then to construct a summary of 10 segments such that some other human could watch the summary and obtain an operational knowledge of the important events that occurred in each stream.

The results of these experiments are discussed next. Section 3.7.1 presents the results of human value judgment elicitation and levels of human agreement; Section 3.7.2 gives detailed experimental results and discussion.

3.7.1 Human Value Judgments

Human match percentages for each of the ten subjects employed in this study are shown in Table 3.2. These figures were computed by removing each subject from the reference set and averaging, for each segment in the test human's summary, how many other humans included that segment. For example, Subject 1's summary for Stream 1 contained, on average, only 6.7% of the segments included in other human-produced summaries. Over all humans and streams, the average human match rate for this corpus is 38.6%, which can be compared to the match rate of a random summarization strategy, 1.2%, or a temporally uniform sampling of segments, 13.5%.

It is additionally worthwhile to examine the distribution of matches for each stream. Figures 3.13 - 3.16 show, for each segment, how many humans included that segment in their summary. As expected, the distribution of segments included in summaries is quite peaky.

3.7.2 Experimental Results

Given the human judgments shown above, 8 different audio-based AARG systems were tested, along with 4 different video-based AARG systems. Audio systems tested included MEL, MFCC, MEL+SPEC, and MFCC+SPEC feature sets, using both the maximum entropy and maximum scene change algorithms. Video systems tested included COLOR and

Subject	Stream				
	Stream 1	Stream 2	Stream 3	Stream 4	Average
h1	6.7%	42.2%	33.3%	31.1%	28.3%
h2	34.4%	43.3%	42.2%	31.1%	37.8%
h3	48.8%	41.1%	54.4%	16.7%	40.3%
h4	27.8%	46.7%	22.2%	30.0%	31.7%
h5	30.0%	47.8%	46.7%	26.7%	37.8%
h6	43.3%	47.8%	46.7%	26.7%	41.1%
h7	46.7%	46.7%	48.9%	43.3%	46.4%
h8	46.7%	53.3%	45.6%	33.3%	44.7%
h9	37.8%	46.7%	36.7%	27.8%	37.2%
h10	46.7%	35.6%	45.6%	35.6%	40.8%
Average	34.9%	45.1%	42.2%	30.2%	38.6%

Table 3.2: Human AARG Match Percentages Per Stream

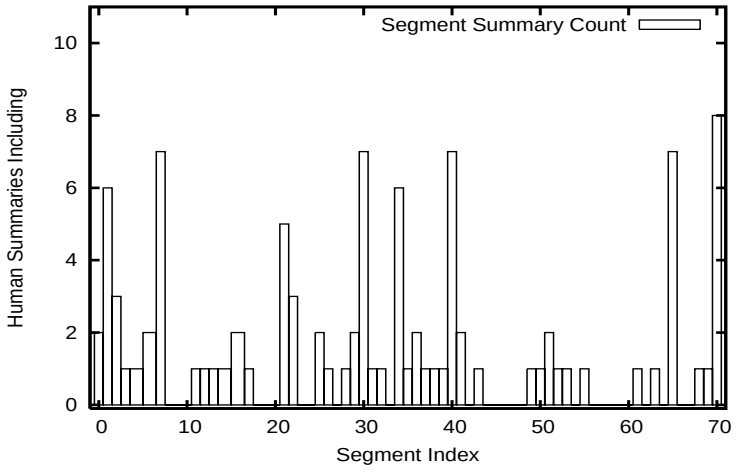

Figure 3.13: Human AARG Summary Segment Counts, Stream 1

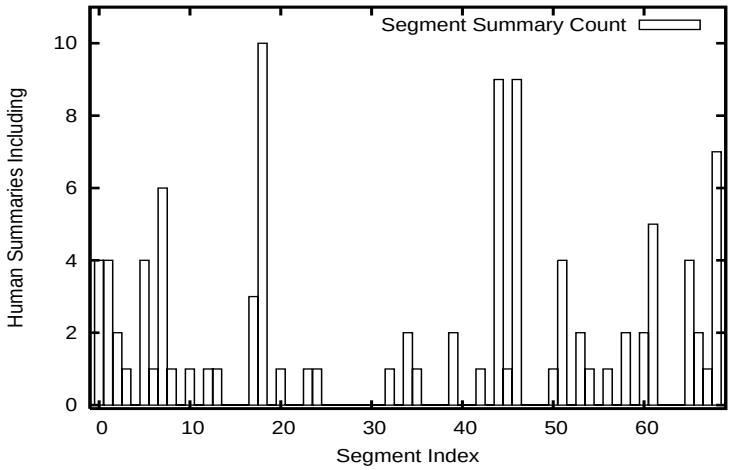

Figure 3.14: Human AARG Summary Segment Counts, Stream 2

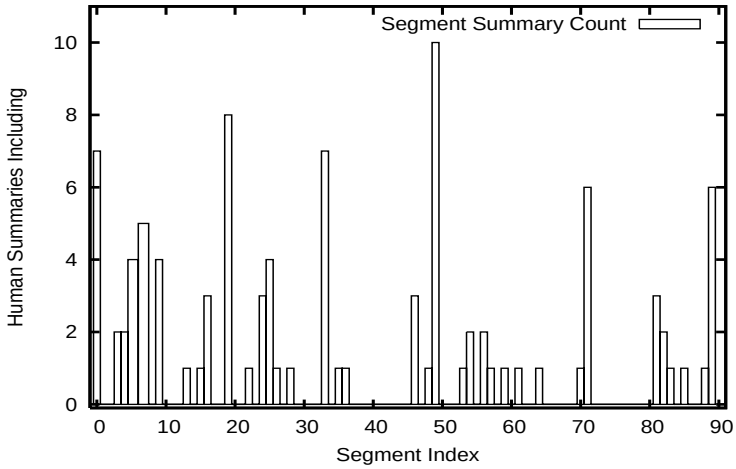

Figure 3.15: Human AARG Summary Segment Counts, Stream 3

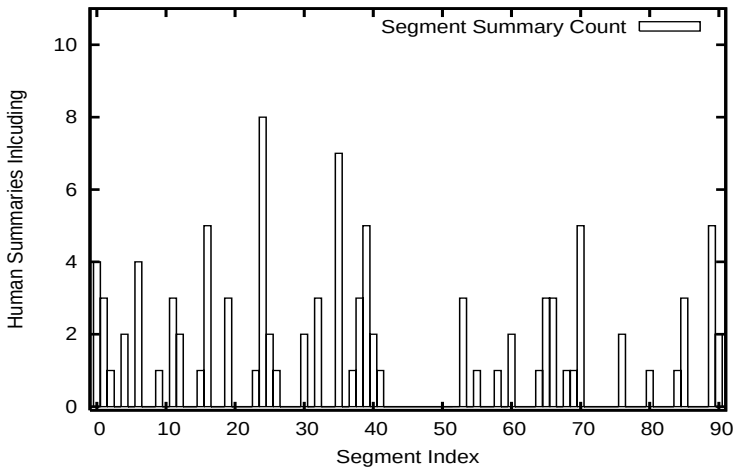

Figure 3.16: Human AARG Summary Segment Counts, Stream 4

MOTION features using both the maximum entropy and maximum scene change algorithms. Note that, since the MOTION features only had a frame rate of 3 per second, only one scene change value per segment was computed, while all other features used five values. Audio results are shown in Tables 3.3 and 3.4; video results in Table 3.5. Both sets of results are shown with match rates for random summarization and temporally uniform summarization for comparison.

There are several results of interest in the audio experiments. First, note that in all cases, the scene change metric appears to perform better than the maximum entropy metric. This result seems to imply that humans cannot or do not keep enough global information in mind to reproduce a summary that is optimal in an information-theoretic sense; rather, they look for highly informative regions (i.e., boundaries) and choose those, independent of other choices. Second, the best-performing system, based on the MEL+SPEC feature set, achieves a match rate of 27%, which is more than halfway between temporal random sampling and average human performance, a reasonable result given the simplicity of the approach. Finally, while adding spectral summary features to the MEL feature set appears

Stream	Feature				Random Gist	Uniform Gist
	MEL		MFCC			
	Gist Type		Gist Type			
	Max Entropy	Scene Change	Max Entropy	Scene Change		
Stream1	18.0%	21.0%	13.0%	16.0%	1.4%	7.0%
Stream2	9.0%	32.0%	9.0%	4.0%	1.4%	27.0%
Stream3	12.0%	27.0%	5.0%	33.0%	1.1%	8.0%
Stream4	14.0%	25.0%	5.0%	10.0%	1.1%	12.0%
Average	13.2%	26.2%	8.0%	15.7%	1.2%	13.5%

Table 3.3: Audio AARG BVA Match Rates per Stream: Basic Features

Stream	Feature				Random Gist	Uniform Gist
	MEL+SPEC		MFCC+SPEC			
	Gist Type		Gist Type			
	Max Entropy	Scene Change	Max Entropy	Scene Change		
Stream1	18.0%	23.0%	15.0%	19.0%	1.4%	7.0%
Stream2	9.0%	32.0%	14.0%	31.0%	1.4%	27.0%
Stream3	19.0%	27.0%	13.0%	34.0%	1.1%	8.0%
Stream4	11.0%	26.0%	14.0%	19.0%	1.1%	12.0%
Average	14.2%	27.0%	14.0%	25.7%	1.2%	13.5%

Table 3.4: Audio AARG BVA Match Rates per Stream: Augmented Features

Stream	Feature				Random Gist	Uniform Gist
	COLOR		MOTION			
	Gist Type		Gist Type			
	Max Entropy	Scene Change	Max Entropy	Scene Change		
Stream1	14.0%	20.0%	13.0%	14.0%	1.4%	7.0%
Stream2	24.0%	18.0%	6.0%	17.0%	1.4%	27.0%
Stream3	4.0%	9.0%	12.0%	11.0%	1.1%	8.0%
Stream4	2.0%	18.0%	1.0%	6.0%	1.1%	12.0%
Average	11.0%	16.2%	8.0%	12.0%	1.2%	13.5%

Table 3.5: Video AARG BVA Match Rates per Stream

to help only a little, adding these features to the MFCC feature set helps a lot, improving performance using the scene change algorithm from 15.7% to 25.7%. This result suggests that while the spectral summary features carry a lot of information that could be useful in summarization applications, it is largely redundant given a spectral feature representation like the melscale filterbank.

The video results are similar to the audio results in that the scene change algorithm outperforms the maximum entropy algorithm, however, performance overall is worse than for audio. The best system achieves a match rate of only 16.2%, which is not much better than temporally uniform sampling, and 10% worse than the best audio system.

The final experiment conducted in the AARG evaluation was to restrict scoring to those segments on which humans demonstrated high levels of agreement, under the intuitive assumption that it is more important to match human judgment when it is largely uniform than when human opinions diverge. Table 3.6 shows the effects of agreement restriction on the best audio system, MEL+SPEC using the scene change algorithm. As anticipated, performance increases with the degree of human agreement to some degree, reaching a

Stream	Human Agreement Level							
	1	2	3	4	5	6	7	8
Stream 1	23.0%	26.7%	23.7%	26.4%	26.4%	29.2%	22.2%	100.0%
Stream 2	32.0%	37.3%	39.1%	40.9%	50.0%	56.1%	48.6%	35.7%
Stream 3	27.0%	31.3%	30.1%	31.1%	38.8%	43.2%	21.9%	0.0%
Stream 4	26.0%	29.1%	32.8%	39.5%	37.1%	53.3%	53.3%	100.0%
Average	27.0%	31.2%	31.7%	34.5%	37.7%	43.2%	33.9%	41.9%
Sample Size	318	81	53	37	27	20	15	7

Table 3.6: AARG BVA match rates at selected human agreement levels

peak of 43.2% on segments included in at least 6 human summaries. Performance falls off above this point; however, the sample size above this level is too small to make general statements.

3.8 Application: Stream-of-Interest Selection for Online Aggregate Remote Sensing

The goal of a stream-of-interest selection application is, given a set of streams presented in real time, to select the one stream that is most informative in some sense. In this thesis, such applications are referred to as online aggregate remote sensing (OARS) applications. These applications are characterized by the presence of multiple actors, possibly not co-located, collaborating in the real world to perform some task and guided by a central authority whose job it is to monitor the field actors and update the task plan. In case the actors are able to transmit live video to the central authority, it is important to filter out the uninteresting information and focus on the streams likely to yield useful information. This application

lends itself well to the BVA approach.

There are two ways in which a stream could be considered informative in an information theoretic sense in OARS applications. First, a stream at a given time segment can contain information that is different from the other streams. This feature is termed *uniqueness*. Second, a stream at a given time segment can contain information that is different from *that* stream in *previous* time segments. This feature is termed *novelty*. In practice, novelty can be computed either as a comparison between the current segment for a given stream and past segments for that stream, or as a measure of the strength of scene change within the current segment. These two types of novelty are referred to as *historical* novelty and *scene change* novelty. In this evaluation, the NSRE metric was used to measure uniqueness and both forms of novelty.

As with AARG, experiments were carried out by first eliciting human value judgments and then comparing BVA judgments against this standard. Splitting the streams into 10-second segments was performed exactly as per the AARG experiments, except that since the task is to select one stream of many, the number of segments had to be clipped at 69, the length of the shortest stream. The four streams were aligned in time and displayed in four-up format using RealPlayer and the SMIL markup language. This four-up display is shown in Figure 3.17. A rather simple value judgment tool was constructed for this task, as shown in Figure 3.18. This tool allowed subjects to view each time segment's streams either in parallel (as would occur in real-world deployment) or in sequence. The subjects were allowed to view each segment as many times as necessary in order to make a value judgment and were told that their goal was to select the one stream in each segment that was most important with the task of maintaining situational awareness over all streams in mind.

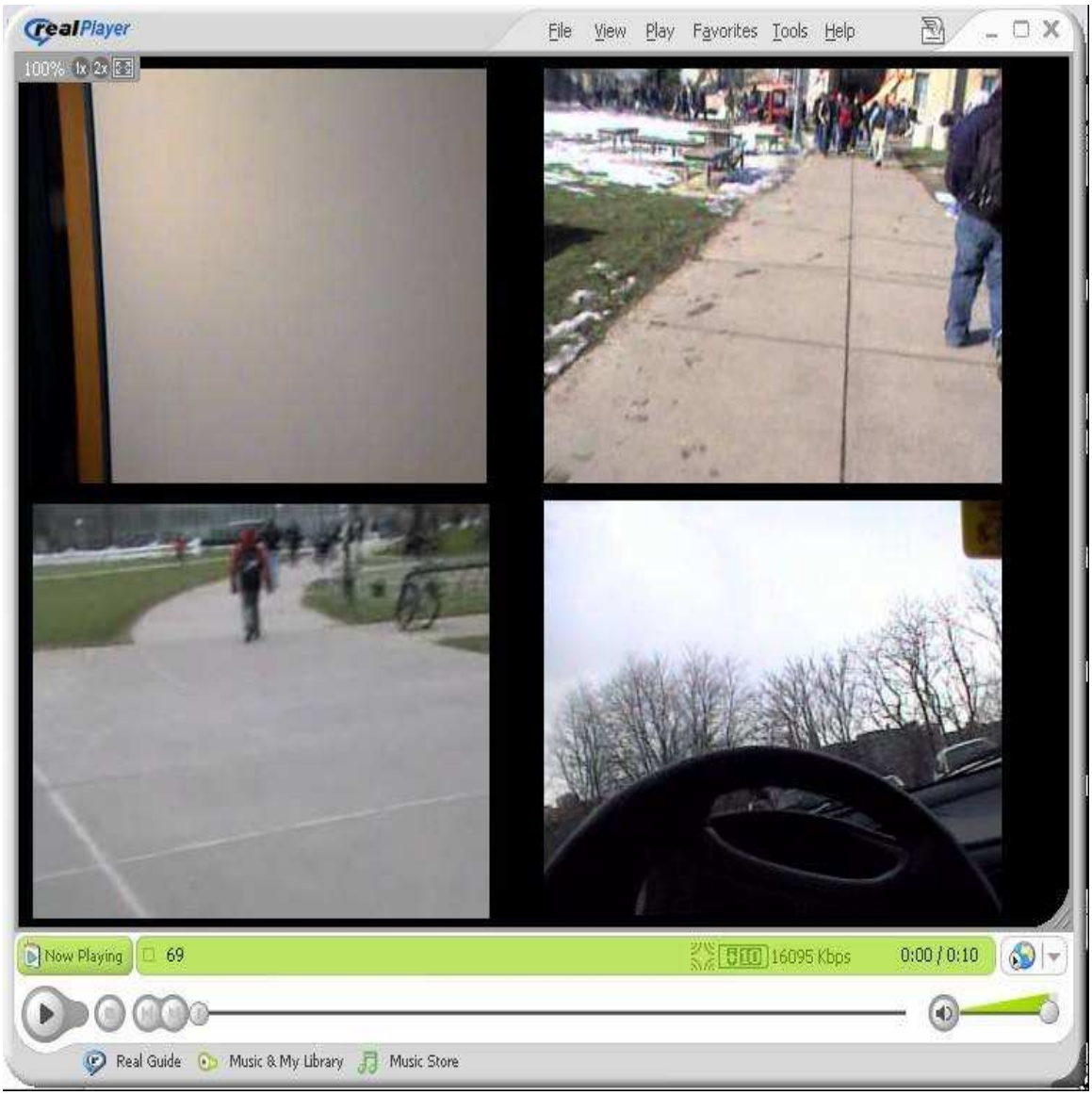

Figure 3.17: The Four-Window OARS Judgment Display

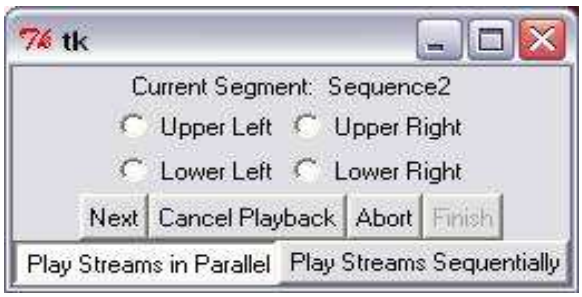

Figure 3.18: The OARS Value Elicitation Tool

3.8.1 Human Value Judgments

In many cases, there was little agreement among humans about which streams were most important. In fact, there were many segments in which not much of anything was happening in *any* stream; in those cases, it appeared that subjects simply chose a stream at random or according to some esoteric preference. However, there were a number of segments where agreement was significant. These segments were mostly those which contained clearly valuable information; e.g., the segment where the author ordered his lunch or started pumping gas. Human agreement levels were measures by computing the Shannon redundancy (see Equation 2.3). A value of 1 indicates unanimity, while a value of 0 indicates a uniform distribution of human responses. Human agreement levels calculated using Shannon redundancy are shown in Figure 3.19. The average agreement level was 0.35; in 18 segments the agreement level was 0.5 or greater.

As with the AARG evaluation, match rates were computed for each human subject. These rates are shown in Table 3.7. Even the best-performing human does not achieve a match rate of 50%; the average human match rate is 41%. This figure is only 16% better than the chance result of 25%.

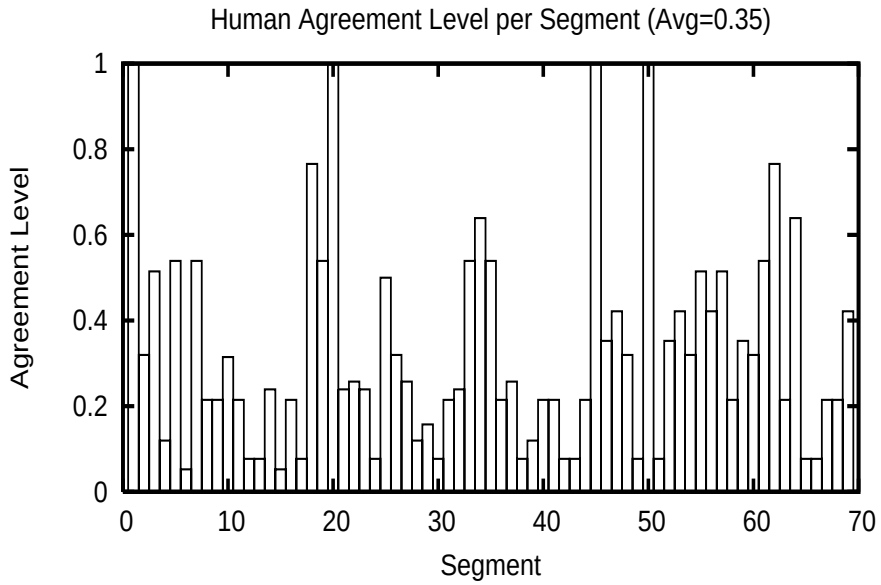

Figure 3.19: Agreement Coefficients Among Human Subjects per Segment

Human Performance			
Subject	Match Rate	Subject	Match Rate
h1	42.19%	h6	33.97%
h2	46.85%	h7	39.45%
h3	42.35%	h8	42.99%
h4	39.29%	h9	41.22%
h5	42.02%	h10	39.61%
Average			40.99%

Table 3.7: Human match rates per subject

Novelty Weight	Feature			
	MEL+SPEC		MFCC+SPEC	
	Novelty Type		Novelty Type	
	History	Scene Change	History	Scene Change
1	23.53%	36.32%	25.44%	33.08%
0.75	25.44%	32.94%	25.58%	30.00%
0.5	23.23%	26.32%	24.11%	20.14%
0.25	20.44%	23.97%	22.05%	20.73%
0	23.08%		20.73%	

Table 3.8: Audio OARS BVA match rates, all test conditions

3.8.2 Experimental Results

In this study, 18 different audio-based BVA systems and 14 different video-based BVA systems were tested. For the MEL+SPEC, MFCC+SPEC, and COLOR feature sets, five different linear weightings of novelty and uniqueness were tested, and for those experiments in which novelty is given a non-zero weight, historical novelty and scene change novelty were evaluated separately. For the MOTION feature, only the historical novelty variant was evaluated. Audio results are shown in Table 3.8; video results in Table 3.9.

Table 3.8 reveals that the best audio-based BVA system achieved a match rate of 36.3%, which is only 5% worse than the average human subject's match rate and 12% better than chance. As with AARG, the feature set based on mel spectra outperformed the MFCC-based feature set, but in this application the difference is not as pronounced; in fact, when using historical as opposed to scene change novelty, MFCCs performed slightly better. Interestingly, though, for both the MFCC+SPEC and MEL+SPEC feature sets, scene change novelty outperformed historical novelty; further, optimal performance was achieved when uniqueness was not considered at all. As with the AARG results above, this result can be

Novelty Weight	Feature		
	COLOR		MOTION
	Novelty Type		Novelty Type
	History	Scene Change	History
1	26.61%	16.91%	21.76%
0.75	24.41%	23.38%	18.38%
0.5	26.03%	23.38%	15.58%
0.25	22.94%	23.38%	15.29%
0	23.38%		15.00%

Table 3.9: Video OARS BVA match rates, all test conditions

interpreted as a human preference for local scene boundaries when searching for relevant information.

Table 3.9 shows that overall, performance when using visual features was not as good as when using audio features, and was in fact very close to chance. Like the AARG task, this result can be taken as evidence that audio may be superior to video for applications where BVA can be used. In this set of results, however, the same relationships between historical and scene change novelty did not hold; the optimal result was achieved by a system using historical novelty.

The final experiment conducted in the OARS evaluation, as with the AARG evaluation, was to restrict scoring to those segments on which humans demonstrated some high level of agreement. Table 3.10 shows the effects of agreement restriction on match rate. Apart from the MOTION feature set, all systems show steady improvement with increasing human agreement level; the optimal result of 61.1% match rate occurs at a human agreement level of 0.5. Above this level, there are too few candidate segments to draw any conclusions; however, the trend, particularly using the MEL+SPEC feature set, is clear.

System	Agreement Level				
	0	0.25	0.5	0.75	1.0
MEL+SPEC	36.32%	48.78%	61.11%	48.33%	50.00%
MFCC+SPEC	33.08%	39.09%	39.44%	35.00%	50.00%
COLOR	26.61%	28.18%	33.33%	22.78%	50.00%
MOTION	21.76%	25.00%	16.67%	21.67%	22.42%
Sample Size	70	34	18	6	4

Table 3.10: OARS BVA match rates at selected human agreement levels

3.9 Chapter Summary

This chapter presented a theory of auditory blind value assignment for multimedia summarization and stream-of-interest selection based on the principle of redundancy reduction. It contained a discussion of two experiments designed to evaluate the theory on a summarization application (AARG) and a stream-of-interest application (OARS) by comparing automatic value judgments with human value judgments on multimedia data.

In the AARG evaluation, the auditory BVA method achieved a human-judgment match rate of 27%, compared to 13% for temporal uniform sampling and 38% for the average human in the study. Restricting the match percentage calculations to those segments on which humans exhibited strong agreement resulted in improved performance. Specifically, when only those segments selected by 60% or more of the subjects were considered, the auditory BVA system's match rate improved to 43%.

In the OARS evaluation, the auditory BVA method achieved a human-judgment match rate of 36%, compared to 25% for chance and 41% for the average human in the study. As with AARG, restricting the match percentage calculations to those segments on which humans exhibited strong agreement resulted in improved performance. Specifically, when

only those segments with a human agreement coefficient of 0.5 or higher were considered, the auditory BVA system's match rate improved to 61%.

For both AARG and OARS systems, melscale spectra proved superior to MFCCs and video features employed for comparison. More interestingly, using strength of scene change within each segment was a much more reliable indicator of human preference than either maximizing summary entropy in the AARG case or maximizing dissimilarity to other segments in the OARS case. This is an interesting result which seems to indicate that local measures of interestingness were much more important to humans trying to complete value assignment tasks than were more global measures. It remains an open question whether local or global measures result in summaries which are more *useful* to human users.

These experiments demonstrate that auditory BVA can be an effective method for both fast exploratory analysis of unstructured multimedia data and for selecting streams of interest in real-time remote awareness applications. Coupled with the modest processing requirements of audio compared to video, these results are a strong argument in favor of blind, audio-first processing of multimedia in advance of more detailed analysis.

Chapter 4

Auditory Environment Recognition for User State Modeling

4.1 Chapter Overview

This chapter describes theoretical and practical work on auditory environment recognition for user state modeling. The discussion begins with the reasons that environment is a potentially useful aspect of user state to consider for certain applications, and moves on to prior work in environment recognition from audio. After presenting two possible approaches to environment modeling, one based on GMM/HMMs, the other on optimal coding, I present experimental results on a large corpus of environmental data including comparisons to human performance.

4.2 Environment and User State

Environment is an important aspect of user state. Human activities and social norms are strongly interrelated with environment. For example, the suite of activities typically pursued while in an office is radically different from those typically pursued in a park or automobile. As these activities inform the willingness of humans to engage in other attention-demanding tasks, like answering a cellular telephone or looking up information, environment is a useful source of information which can be exploited by context-aware mobile applications in order to make their behavior more sensitive to human social preferences. Using the cellular telephone example, it is typically considered socially inappropriate to accept a telephone call while in a theater or meeting unless the call is urgent. Likewise, accepting a call while driving could be distracting, while attempting to accept a call at a loud construction site might be futile. Aside from avoidance of inappropriate human-machine interactions, context-aware systems might be able to learn associations between environments and information-seeking behaviors. For example, a user might often look up stock quotes while riding the bus or train to work; a system which learns to associate this behavior with the bus or train environment might be able to anticipate the user's needs and appear to be more responsive. In order to achieve this kind of context-aware behavior, though, mobile devices need some means of *detecting* environmental types.

4.3 Prior Work

There has been a moderately-sized body of prior work on recognition of specific locales, as opposed to environments, from audio data in both supervised and unsupervised settings. As these two problems are intuitively similar, it is worthwhile to explore the techniques previously used for locale recognition.

Clarkson used both audio and video data, clustered with multilayered HMMs, to record,

segment, and access long-term recordings of day-to-day life; see [17], [20], [18], [19], and [16]. The earliest of these works focused mainly on audio, and found that it was difficult to distinguish between locales which sounded similar using this modality; for example, it was hard to tell a lab lounge from the author's office. For this reason, Clarkson's later work added features from additional modalities, specifically, visual and orientation features. Note however, that the difficulty in making this kind of distinction makes perfect sense and suggests that the more general problem of environment recognition using only auditory information is possible.

In [30] and [29], Ellis and Lee used auditory information to cluster long personal recordings into coherent scenes, which corresponded more or less to specific locales. In [30], a spectral clustering algorithm due to Ng, Jordan, and Weiss [74] was applied to audio segments produced by a BIC segmentation method; the overall precision of this clustering based on 16 labeled classes was 61%. In [29], an exploration of optimal features for segmenting long personal recordings was made. This study indicated that the best features for this task were minute-long metafeatures based on Bark-scale spectra; specifically, the average log energy, average entropy, and entropy deviation of short-term Bark-scale spectra were most useful.

In [54], Kapoor and Basu demonstrated the use of an auditory version of epitomic image representation (due to Jojic [53]) to model and classify a few basic audio classes, and speculated that this feature representation could be useful for auditory environment recognition.

A coding approach to *visual* scenes due to Oliva, Torralba, and collaborators is related to the auditory coding approach presented in this chapter. This method can be used for high-level analysis of environment or locale and can be found in [102], [103], and [76].

The author performed two pilot studies, reported in [63] and [66]. The first study involved a 6-class test with one recording per environment. The environments tested included office, atrium, car, lecture, street, and CMU campus. Using this corpus, one autoencod-

ing neural network was trained for each environment using 17 MFCCs, spectral centroid, spectral rolloff, zero-crossing rate, and power sampled at a frame rate of 5 Hz. This system achieved an error rate of 1.7% on the test data. By comparison, four human subjects achieved error rates of 21% on a cold test and 12% after being allowed to listen to the training data. Note that since there was only one recording per environment, though, this was in reality another locale recognition system.

The second pilot study used 11 classes; apartment, hallway, elevator, lecture, meeting, office, outside, raining, restaurant, theater, and vehicle. Unlike the previous study, multiple locales per environment were used; however, the corpus collected was unbalanced. Using this corpus, one autoencoding neural network and one GMM were trained per environment; 64 MFCCs plus spectral centroid were compressed using PCA to 35 dimensions at a frame rate of 100 Hz. Both systems achieved error rates of 22%; a hybrid system improved performance to less than 20%. This system, unlike the first pilot system, was in fact an environment recognition system capable of performing some measure of generalization.

More recently, Chu et. al. reported experiments on environment recognition for mobile robots using a variety of audio features and classifiers [15]. They obtained a recognition error of approximately 6% on their dataset using a k-nearest neighbor classifier and a mixture of MFCCs, zero-crossing rate, standard deviation of zero-crossing rate, and standard deviation of spectral rolloff. Their dataset consisted of five environmental types; hallway, cafe, lobby, elevator, and outside. It should be noted, however, that like many studies of this type, there was only one locale studied per environment; in other words, these experiments described a locale recognition task rather than the much more difficult general environment recognition task.

4.4 Gestalt Environment Modeling

As prior work demonstrates, it is possible in principle to recognize environment types from acoustic data. This is the case both because different environments tend to contain characteristic events and because they tend to have characteristic gross background features like noisiness, reverberation, and the like. Taking both foreground and background features together, different environments often have characteristic sound fields. Rather than attempt to consider the foreground and background characteristics of a sound field separately, which would require some form of source separation, one might instead view the entire sound field as an entity to be modeled.

This kind of gross or Gestalt modeling of the environment can be done using conventional generative or discriminative methods, e.g., GMMs/HMMs or MLPs. The perceptual principle of redundancy reduction, however, suggests that an alternate approach based on environment-specific optimal coding might be possible. The generative approach and the optimal coding approach are discussed below.

4.4.1 A GMM/HMM Approach

The GMM/HMM approach to environment recognition involves the use of rather standard machine learning techniques. For each environmental class E_i, an acoustic model M_i is trained using the Expectation Maximization (EM) algorithm. [27]. During testing, for each data segment S_j, the hypothesis is taken as $\arg_{E_i}\max P(S_j|M_i)$.

The main implementation issues of this approach are issues of model structure. If GMMs are used, the main issue is how to choose the appropriate number of Gaussians. If HMMs are used, the main issues are how to select a topology and how many Gaussians to use in each state. For GMMs, the BIC, given in Equation 3.4 is a useful measure for model order selection. One can iteratively test the BIC with an increasing number of Gaussians and select the model order which maximizes the criterion. For HMMs, several

topology induction methods have been proposed as discussed in Section 2.6. In this work, the k-variable k-means algorithm due to Reyes-Gomez and Ellis [83] (a modification of an algorithm used by Zhang and Kuo in [116]) is used. This algorithm, which is in the leader-follower class of clustering algorithms, is defined as shown in Figure 4.1.

k-Variable k-Means

1 Given: F, a t-length sequence of n-dimensional feature vectors,
  z, the minimum number of samples per model,
  θ, the generality threshold; $0.5 <= \theta <= 1$.
2 Compute: m, the average distance between vectors in F,
  s, the standard deviation of distances between vectors in F.
3 $\theta_m \leftarrow m - s\theta$
4 $\theta_s \leftarrow m + s\theta$
5 Initialize model c_0 with frame having highest norm.
6 **repeat** until $V = \sum_{jk}(x_{jk} - c_k)^2$ is minimized:
7 **for** each unassigned frame F_i
8 $d \leftarrow \min(d_{ij}(F_i, c_j))$.
9 **if** $d < \theta_m$
10 **then** add F_i to c_j.
11 **elseif** $d > \theta_s$
12 **then** make new cluster c_α with F_i as center.
13 Remove clusters with fewer than z samples.
14 Assign all unclassified frames to closest cluster.

Figure 4.1: The k-variable k-means algorithm

4.4.2 A Redundancy Reduction Approach

The redundancy reduction approach to environment recognition rests on the fact that the optimal code for a suite of signals depends only on the statistics of that suite of signals. If one suite S_x varies substantially from some other suite S_y, then the optimal codes for these suites will also differ substantially.

If one posits that different environments in fact contain different kinds of acoustic signals, then this difference in optimal codes can be used to discriminate between environments in the following way. For each environment χ, an optimal coding matrix C_χ can be derived. Applying W_χ on some set of n-dimensional feature vectors X drawn from χ yields a coded representation $H(X)_\chi$. A decoding matrix D_χ can be derived from X and C_χ in the following way:

$$D_\chi = (H^T H)^{-1} H^T X, \tag{4.1}$$

$$= ((XC_\chi)^T (XC_\chi)^T)^{-1} (XC_\chi)^T X. \tag{4.2}$$

The decoding matrix can then be applied to $H(X)_\chi$ to yield an estimate of the original input, $\hat{X}_\chi$. The difference between X and $\hat{X}_\chi$, computed as:

$$\Delta(X, \hat{X}_\chi) = \sum_{i=1}^{|X|} \sum_{j=1}^{n} (X(i,j) - \hat{X}_\chi(i,j))^2. \tag{4.3}$$

and is referred to as the coding error of X given χ. If X was in fact drawn from environment χ, one would expect that $\Delta(X, \hat{X}_\chi)$ would be smaller than the coding error of X given the optimal coding and decoding matrices from some other environment ψ; i.e., $\Delta(X, \hat{X}_\chi) < \Delta(X, \hat{(X)}_\psi)$. Given this expectation, one can construct optimal coding and decoding matrices for each environment α and, during testing, take as the hypothesis $\arg_\alpha \min \Delta(X, \hat{X}_\alpha)$.

As with the GMM/HMM approach, there are engineering issues to address. First is the order of the coded model. This order must be smaller than the dimensionality if the original feature vector [1]; one would assume that the more features the coded signal has, the more faithful the reconstruction of the original signal would be. Conversely, the closer the dimensionality of the coded signal to the original signal, one would assume that the coding matrix would approach the unit matrix, reducing discriminability. The second issue to address is the method for deriving the coding matrix. PCA and ICA are obvious candidates; however, a numerical method using autoencoding multilayer perceptrons (MLPs) is also a possibility. Autoencoding MLPs are those trained using the input as the target output; when the mean squared error function is used as the training criterion, the hidden weights will approximate the PCA solution [26]. More specifically, the MLP approach will learn the PCA solution up to a rotation. While analytical PCA is faster to train and always gives an exact solution, the MLP approach has the advantage of being able to converge on the specific rotation of the PCA solution which is optimal in terms of coding error with a given number of coding units.

It is also possible to extend the optimal coding approach by employing a mixture of coders. In this approach, data are fragmented and multiple coders are trained in the hope that each group of coding elements will model a specific spatial cluster of the given class. In this work, trees of MLP autoencoders are used to test the mixture of coders approach.

4.5 Experimental Evaluation

In order to evaluate the validity of the proposed methods, a large representative database covering typical environments encountered by the target users in the CHIL Connector scenario [25] was developed. Environments such as office, lecture, and meeting were left out, as the assumption of the CHIL project is that these environments would be instrumented

[1] Unless we are using an overcomplete representation, e.g., per Lewicki and Sejnowski [59]

in such a way as to make environment recognition redundant; the CHIL-equipped space would simply notify the Connector device of the user's locale.

This section describes in detail the data collected and used for the environment recognition evaluation, as well as results of a small study on human performance on the acoustic environment recognition task designed to provide a baseline against which to judge machine performance.

4.5.1 Data Collection

The database used in this work consists of approximately 20 hours of audio data recorded in 14 different kinds of environments in 10 different countries on 4 continents [2]. The data were recorded in ten-minute chunks using a Sony minidisc recorder with a Sony ECM-717 stereo microphone and converted to mono 16-bit, 16 kHz raw format. The data collector was instructed to remain stationary during recording, and to always use identical microphone placement and recording settings.

From this database, nine environments were selected for study. These environments are airport, bus, gallery, park, plaza, restaurant, street, train, and train platform. These environments were selected to be representative of the environments encountered in the CHIL Connector scenario. Most are self-explanatory. "Gallery" refers to any crowded *indoor* space not covered by the other environments, e.g., a mall. "Plaza" refers to any crowded *outdoor* space not covered by the other environments; e.g., a city square or piazza with no significant vehicle traffic. "Train platform" refers to the actual area with train tracks, where passengers board and disembark from subway cars or high-speed trains. "Train" refers to subways, high-speed trains, and street trolleys.

From each environment, 7 recordings were selected at random, and divided into two pools. The first pool consisted of 6 recordings from each environment and was labeled the

[2] Thanks again to Kornel Laskowski for creating this database during his travels in 2004 and 2005.

"seen" pool. The second pool, made up of the remaining recording from each environment, was labeled the "unseen" pool. Each 10-minute recording was then divided into 120 5-second segments; the test set was constructed from the 120 unseen segments, plus 120 segments from the seen pool. The remaining segments were assigned to the training set. The training set thus consisted of a total of 5400 5-second segments, and the test set of 2160 5-second segments, evenly split between seen and unseen conditions. Additionally, 108 segments from the test set were used to evaluate human performance. This human evaluation subset consisted of 12 segments per environment, 6 from the seen pool and 6 from the unseen pool. This data assignment was the same one used for the 2006 NIST CLEAR evaluation [64].

4.5.2 Human Performance

Establishing a human performance level in this research serves two purposes. First, human performance serves as a benchmark for machine listening algorithms. Traditionally, human performance has been used as a gold-standard for progress on perceptual tasks like ASR, speaker identification, and image analysis. Second, one can examine the types of errors that humans make, and, if systematic errors are found, these can be compared to the types of errors made by machine listeners in hopes of gaining some insight into the differences between human and machine perception.

In order to evaluate human performance, the human subset of the test set, described above, was used. Ten human subjects were first briefed on the nature of the task and the environments in the test, and then given a 108-question multiple choice test. The subjects were not told that the corpus was balanced, nor were they told how many different recordings were present for any environment. They were not given access to previous answers, or allowed to change previous answers; they were, however, allowed to listen to any segment as many times as they liked before giving an answer.

Subject	Total Error	Seen Error	Unseen Error
1	72.22%	70.37%	74.07%
2	75.00%	75.92%	74.07%
3	70.37%	75.92%	64.81%
4	75.92%	66.67%	85.15%
5	76.85%	81.48%	72.22%
6	70.37%	72.22%	68.51%
7	73.15%	77.78%	68.51%
8	76.85%	79.62%	74.07%
9	76.85%	74.07%	79.62%
10	73.70%	74.44%	68.51%
Average	73.70%	74.44%	72.96%

Table 4.1: Human Performance on Environment Recognition Task

A summary of human performance is shown in Table 4.1. Performance was on average poor, and the difference between the best performance and the worst was only 6.5% absolute, a fairly small margin given the magnitude of the errors. The average performance was only 15% better than chance.

The types of errors made by humans are illuminating. Significant confusions existed between several pairs of classes that would seem to be intuitively close acoustically. Bus and train, train platform and street, and restaurant and gallery are members of this class of confusion. Particularly interesting is the confusion between park and plaza; here, it seems, human listeners latched onto a particular cue which is frequent in both classes — the sound of birds chirping — and interpreted this as a strong park cue. Finally, there are a number of other confusions in which it seems that humans could not even tell whether the environment was indoors or outdoors — airport and plaza, gallery and street, and restaurant

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	12	3	22	14	21	16	14	5	13	10.00%
bus	1	24	3	1	3	20	11	42	15	20.00%
galry	19	1	19	2	10	14	20	3	32	15.83%
park	0	7	7	91	6	1	4	4	0	75.83%
plaza	4	3	21	44	20	8	10	0	10	16.67%
restr	12	1	27	4	27	39	5	3	2	32.50%
strt	6	4	16	10	38	9	23	5	9	19.17%
trn	14	27	16	1	5	8	10	34	5	28.33%
platf	11	11	20	4	11	7	23	11	22	18.33%
PRECISION	15.19%	29.63%	12.58%	53.22%	14.18%	31.97%	19.17%	31.78%	20.37%	

Table 4.2: Environmental Confusions, Precision, and Recall: Human Subjects

and plaza were all in this class. This confusion would seem to indicate that humans are in fact listening for specific sound cues rather than trying to consider the characteristics of the entire sound field, such as noisiness or reverberation, to make their decisions. This is an appealing hypothesis, as it is often difficult for untrained humans to describe a sound field as anything other than a collection of specific sounds. The entire confusion matrix is shown in Table 4.2. Per-class F-scores are shown in Table 4.3, while the top confusions by percentage of response are shown in Table 4.4.

As the confusion matrix and F_1 score table show, no class was truly easy for humans, though their best performance came on the park class, with an F_1 score of 62.54. Other *relatively* easy classes for humans included park, restaurant, train, and bus; conversely, plaza, gallery and airport were the hardest. This result is intuitively appealing in that the classes one might think of as the most distinctive in terms of sound field are the easiest for humans to distinguish. Likewise, the more nebulous classes are the hardest. That airport should be the hardest class for humans is surprising at first blush; however, when one considers all the different sub-environmental types present in an airport; e.g., checkin areas, departure gates, skymalls, baggage claim, passport control, customs, etc., it is much easier to understand why humans should have such a hard time with the class.

The top 10 human confusions are dominated by the train / bus pair; confusions between

Environment	F_1 Score
park	62.54
restaurant	32.23
train	29.95
bus	23.88
train_platform	19.29
street	19.17
plaza	15.32
gallery	14.01
airport	12.06

Table 4.3: Human F_1 Scores

Rank	Ref	Hyp	Pct	Rank	Ref	Hyp	Pct
1	Plaza	Park	4.07%	5T	Restaurant	Gallery	2.5%
2	Bus	Train	3.88%	7	Platform	Street	2.12%
3	Street	Plaza	3.51%	8	Airport	Gallery	2.04%
4	Gallery	Platform	2.96%	9T	Plaza	Gallery	1.94%
5T	Train	Bus	2.5%	9T	Airport	Plaza	1.94%

Table 4.4: Top 10 Human Confusions as Percentage of Total Answers

these two classes account for 6.5% of the total responses, or nearly 9% of the total errors made. Except for platform / street, the remainder of the top 10 errors were confusions between the three most ill-defined classes: airport, gallery, and plaza.

4.6 Experimental Results: GMM/HMM Approach

Here experiments using the GMM/HMM approach are described, including studies on basic feature selection, GMM model order, and HMM topology selection. After discussing each experiment in turn in Sections 4.6.1 - 4.6.6, a summary of these results and error analysis are presented in Section 4.6.7.

4.6.1 GMM Feature Selection

The first experiment was on feature selection using diagonal-covariance GMMs with 100 Gaussians per class. Four feature sets were considered, all computed at 100 frames per second:

MEL A set of 14 melscale spectra.

MFCC A set of 14 MFCCs derived from 64 melscale spectra.

MEL+ A set of 11 melscale spectra, plus SNR, spectral centroid, and spectral energy diffusion.

MFCC+ A set of 11 MFCCs derived from 64 melscale spectra plus SNR, spectral centroid, and spectral energy diffusion.

The spectral summary features SNR, spectral centroid, and spectral energy diffusion were chosen for their presumed applicability to the environment recognition problem; that is, each of these features should vary across environments in predictable ways. All these

Feature Set	Total Error	Seen Error	Unseen Error
MFCC	18.93%	8.24%	29.63%
MEL	41.04%	38.61%	43.42%
MFCC+	17.36%	8.33%	26.39%
MEL+	33.65%	31.85%	35.46%

Table 4.5: GMM Feature Selection — 100 Gaussians Per Class

features were normalized to zero mean and unity variance as computed over a uniform sampling of the entire training corpus.

In order to evaluate these features, a set of 100-Gaussian GMMs were trained, one per environmental class. These GMMs were initialized using the neural gas algorithm, a soft variant of the k-means algorithm [69], and trained for 10 iterations using EM. Results are shown in in Table 4.5.

Several results in this table are noteworthy. First, MFCC-based feature sets performed significantly better than MEL-based feature sets, which seems to indicate that the DCT as an optimizing transform of spectral coefficients is useful for sound field modeling. Second, the augmented feature sets MEL+ and MFCC+ both achieved better performance than the unaugmented feature sets; in the MFCC case by 8% relative and in the MEL case by 18% relative. This result indicates that using the spectral summary features SNR, centroid, and diffusion is better than having three more basic coefficients, meaning that these features are indeed useful for environment recognition. Third, the performance improvement for MFCC+ over MFCC is entirely due to an 11% relative improvement on the unseen data, meaning that the spectral summary features are in some way more characteristic of environments than basic MFCCs. Fourth, there is a considerable performance gap between the seen locales and the unseen locales. While the performance on unseen locales is still much better than chance (an 88.9% error rate) and hence potentially useful, it is clear that actu-

Frame Rate	Smoothing	Total Error	Seen Error	Heldout Error
100 fps	no	17.36%	8.33%	26.39%
100 fps	yes	21.01%	4.16%	37.87%
10 fps	no	26.80%	12.50%	41.11%

Table 4.6: GMM Feature Selection — Effects of reducing short-term temporal structure

ally generalizing to new locales is much more difficult than simply recognizing locales that were seen in training. Finally, note that note that these performances are much better than human performance, a subject that will be explored in more detail in Section 4.8 below.

In addition to this basic feature selection experiment, tests designed to remove the effects of short-term feature variability were also conducted. Some researchers, notably Ellis [29], have used very long analysis windows for environment recognition on the basis of the fact that environments themselves are slow-changing; thus, short-term temporal structures might in fact be misleading. One way to reduce short-term structure is to lower the frame rate. Another is to employ feature smoothing, for example, the scale-space smoothing approach discussed in Chapter 3. Both of these approaches were evaluated using the MFCC+ feature set; note that, since the reduced frame rate approach compresses the set of training examples by an order of magnitude, the number of Gaussians per class is correspondingly reduced from 100 to 10. Further, the smoothing factor σ was set equal to 2 for this work. Results of these experiments are shown in Table 4.6.

Neither smoothing nor reduction of frame rate improves performance overall. However, it is noteworthy that smoothing causes performance on seen locales to *improve* by 50% relative, while performance on unseen locales is degraded by 43% relative. This result would seem to indicate that smoothing somehow impairs the ability to generalize while improving the ability to recall previously seen locales. A similar, though much less dramatic result is seen in the slow frame rate condition. Here, performance on seen locales degrades by

50% relative, but performance on unseen locales degrades even more, by 56% relative. It is likely that the overall worse performance of the slow frame rate system is simply due to fewer parameters. However, the fact that in both experiments, performance degrades more on unseen locales would seem to indicate that while more gross auditory characteristics are important for locale recognition, it is the short-term temporal structures which are important for generalization. This is a somewhat counter-intuitive result. However, insofar as specific sound types are often characteristic of environments rather than locales (i.e., the sound of a train pulling up to a train platform, or birds chirping in a park), it is reasonable.

Note that the choice between specificity and generalization is largely application-dependent. Hence, it is possible that in some applications smoothing should be employed in order to recognize specific locales with high accuracy. However, since the goal of this thesis is to learn something about the properties of environmental sound fields in general, the use of smoothing is not indicated for further experimentation.

4.6.2 GMM Model Order Selection

The next experiment involved varying the number of Gaussians per class using the MFCC+ feature set in order to measure the effect of parameter size on performance. One might expect that, as the number of parameters increases, performance on seen locales would increase, possibly at the expense of performance on unseen locales. To measure this effect, a number of GMM sizes were evaluated. Each environmental class had 300,000 training examples; if each parameter requires 100 samples to adequately estimate, then the maximum number of Gaussians per class is 107 (disregarding mixture weights). Systems using 25 Gaussians per class through 125 Gaussians per class were tested; the 125-Gaussian case was intended to investigate in fact whether 100 samples per parameter were adequate, inadequate, or more than adequate. Results are shown in Table 4.7.

While performance on seen locales does indeed improve with more parameters, so does

Gaussians / Class	Total Error	Seen Error	Heldout Error
25	23.33%	13.05%	33.61%
50	19.54%	10.18%	28.89%
75	17.41%	8.33%	26.48%
100	17.36%	8.33%	26.39%
125	16.99%	8.05%	25.92%

Table 4.7: Effects of GMM Parameter Size

performance on unseen locales. By using 125 Gaussians per class instead of 100, overall performance improves. Though the improvement is mostly due to better locale recognition, generalization does not suffer; in fact, the gap between unseen and seen performance drops monotonically from 20.5% with 25 Gaussians to 17.8% with 125 gaussians. This result is rather straightforward to interpret. First, 100 samples per parameter appears to be more than enough; the 125-Gaussian system had only 82 training examples per parameter. Second, though additional parameters do help locale recognition, they help close the gap between locale recognition and generalization. In other words, the extra models being learned are in fact, at least to some degree, broad environmental models.

4.6.3 GMM Feature Transformations

The MFCC+ feature set, as shown above, achieves reasonable performance levels on the environment recognition task. Here, various "optimal" feature space transformations are tested. Specifically, PCA, ICA, and LDA are evaluated in order to determine whether or not these transforms can be of use for environment recognition. As noted in Chapter 2, PCA and ICA attempt to transform the feature space in such a way that the input features are most faithfully *represented*, while creating output features that are, respectively, either decorrelated or statistically independent. LDA, by contrast, seeks a transformation such that the

Transform	Total Error	Seen Error	Unseen Error
None	16.99%	8.05%	25.92%
PCA	21.71%	10.37%	33.05%
ICA	23.93%	11.29%	36.57%
LDA	20.18%	9.72%	30.65%

Table 4.8: Effects of Feature Transformations, 125 Gaussians per Class

output features are maximally linearly separable given some class labels. For this experiment, output feature space dimensionality was kept at 14; in other words, full transforms were used. The ICA transform was produced with the deflation variant of FastICA [48] with the $\tanh$ nonlinearity. Given the goals of these transformations, it was expected that LDA would improve performance over the baseline, that LDA would perform better than either of the other two methods, and that ICA would perform better than PCA. Results are shown in Table 4.8.

In fact, only one of the expectations of this experiment held; namely, that LDA outperformed PCA and ICA. Conversely, LDA did *not* perform better than the baseline MFCC+ feature set, nor did ICA outperform PCA. Examining the LDA result in more detail, note that most of the degradation comes in the unseen condition. Since it is apparent that the differences between the seen and unseen data are not small, it is reasonable that a linear transformation bent on linear separability in the seen data pool might cause a performance degradation on unseen data. This conjecture does not explain why performance on the seen data is worse. It may be the case that there is little linear separability between sound field classes as a whole. That is, the discriminability between sound field classes may rest in small components of the sound fields, i.e., in individual sound cues. This contention is similar to the one made to explain the failure of smoothing, above. In ASR systems, LDA is typically applied at the level of HMM states, which in practice means context-

Context Size	Total Error	Seen Error	Unseen Error
0 Frames	16.99%	8.05%	25.92%
3 Frames	23.33%	10.00%	36.67%
5 Frames	23.65%	8.98%	38.51%
7 Frames	22.17%	8.42%	35.92%
9 Frames	22.36%	8.24%	36.48%
11 Frames	23.56%	8.14%	38.98%

Table 4.9: Effects of Context+LDA, 125 Gaussians Per Class

dependent sub-phone units. These are very small structures; it is likely that any attempt to introduce LDA at a higher level, e.g., the phone level, might actually hurt, as is the case with environment-level LDA in this experiment. This is not to say that LDA is useless for environment recognition; merely that it cannot be applied at a high level.

4.6.4 GMM Temporal Feature Context

Temporal context is widely used in modern ASR systems, allowing trajectories of features over time to be tracked and modeled. Temporal context has also been used in other general audio modeling applications; see work by Slaney [90], Malkin [66], [67], and Kraft [56], [55] for examples. In this work, consecutive frames in the source feature set are stacked to create much longer feature vectors which are then processed by some dimensionality reducing transformation in order to avoid the problems of a very high dimensional feature set. In this experiment, following the results shown in Section 4.6.3, LDA transforms were learned at the environment level; these transforms reduced the feature space to 14 dimensions. Keeping the number of Gaussians per class fixed at 125, the context width was varied between 3 and 11 frames. Results are shown in Table 4.9.

As these results demonstrate, use of context in the feature space does not help performance, at least when using an LDA transformation. Interestingly, there is a clear trend of improving performance on the seen data from shorter contexts to longer contexts, while this trend is not present in the unseen data. This result would seem to indicate that the feature trajectories learned by LDA improve with length for locale recognition, but that these trajectories are not especially useful for generalization. In fact, they are all worse than single-frame LDA, meaning that if temporal context is useful for environment recognition at all, the feature space is not the proper place to implement it.

4.6.5 HMM Topology Selection

HMMs are widely used in general audio recognition systems and merit investigation for the environment recognition problem. A major issue, however, as noted above, is that it is not clear *a priori* what kind of topologies are appropriate for this task. Reyes-Gomez and Ellis' k-variable k-means algorithm, shown in Figure 4.1 is able to learn HMM topologies from data in a principled way. In this work, a minimum of 100 training examples per parameter were required, which amounts to 2800 training examples per HMM state. Additionally, the BIC was used to determine the number of Gaussians per state rather than fixing the number of Gaussians based on the number of training examples. As a result, the HMM topology selection procedure in this work eliminates many more states than the version described by Reyes-Gomez and Ellis. After selecting the HMM topology for each environment and the number of Gaussians per state, the GMM parameters are initialized using the neural gas algorithm and the HMMs are trained using 10 iterations of Viterbi training. HMM transition parameters were not optimized during training, but left at their initially-observed values from the k-variable k-means procedure. As the transition probabilities are much smaller than the acoustic probabilities, they do not have much impact on Viterbi decoding.

Five different generality thresholds were used to build 5 different HMMs per environ-

Environment	Generality Threshold				
	0.5	0.75	1	1.25	1.5
airpt	33	28	22	18	11
bus	31	35	22	16	11
galry	34	32	22	18	7
park	15	11	7	7	3
plaza	27	24	22	15	9
restr	33	32	28	23	21
strt	35	23	20	16	6
trn	32	28	24	18	5
platf	23	18	19	18	13
Total States	263	231	186	149	86
Total Gaussians	765	848	850	844	834
Gaussians Per State	2.9	3.7	4.5	5.6	9.7

Table 4.10: HMM States Per Environment

mental class using the MFCC+ feature set. The resulting number of states per class, total states per system, total Gaussians per system, and number of Gaussians per state, are shown in Table 4.10.

On average, the number of states per class decreases with the generality threshold, and the number of Gaussians per state increases with the generality threshold. The total number of Gaussians, however, appears to be relatively constant at thresholds 0.75 and higher. All of these HMMs were evaluated on the seen and unseen data sets. Results, together with the best GMM system for comparison, are shown in Table 4.11. Also shown in this table is a hybrid HMM system which consists of a linear interpolation of all k-variable k-means HMM scores.

KVKM Threshold	Total Error	Seen Error	Unseen Error
0.5	19.17%	5.00%	33.33%
0.75	16.80%	5.83%	27.79%
1	15.41%	5.46%	25.37%
1.25	17.13%	5.74%	28.51%
1.5	16.85%	7.13%	26.57%
Hybrid	16.06%	6.02%	26.38%
GMM	16.99%	8.05%	25.92%

Table 4.11: HMM Error Rates

The best HMM system, using a generality threshold of 1, achieved an error rate of 15.41%, some 9% better, relative, than the best GMM system. This improvement comes almost entirely from a 32% relative improvement on the seen data; the improvement on the unseen data is only 2% relative. It is important to note, however, that this HMM system had a total of 850 Gaussians and 186 HMM states for a total of 28,756 parameters; by contrast, the GMM system had a total of 1,125 Gaussians for a total of 32,625 parameters. The HMM system's performance gain comes in spite of having nearly 12% fewer parameters than the GMM system, which is a fairly strong indication that HMMs are a much better tool for environment recognition than GMMs, even if the topologies must be estimated from data. Note also that the single best HMM on seen data is the 0.5 threshold HMM, which has 32,849 parameters, even more than the GMM system. This model appears to have learned to recognize specific locales at the cost of generalizability. Finally, the hybrid system's error rate is smaller than the average component system error, but is not better than the single best system.

System	Params	Total Error	Seen Error	Unseen Error
100-GMM-MFCC	26,100	18.93%	8.24%	29.63%
100-GMM-MFCC+	26,100	17.36%	8.33%	26.39%
125-GMM-MFCC+	32,625	16.99%	8.05%	25.92%
1-HMM-MFCC+	28,756	15.41%	5.46%	25.37%
1-HMM-MFCC+-Train20	28,756	15.18%	5.09%	25.27%

Table 4.12: GMM / HMM Error Rate Summary

4.6.6 Additional Experiments

Training the 1-threshold HMM for an additional 10 Viterbi iterations yielded a small improvement to 15.18% total error, with 5.09% error on the seen data and 25.27% on the unseen data. A final experiment involved pruning HMM transitions with less than 1% probability; this experiment removed 2000 of the 4106 transitions in the model, but resulted in a performance degradation to 15.78% total error, with 5.27% on the seen data and 26.29% on the unseen data.

4.6.7 GMM/HMM Result Summary

Table 4.12 shows a summary of results for the GMM/HMM system. The best result, 15.18% total error with 5.09% seen error and 25.27% unseen error, was acheived by 20 iterations of Viterbi training on a set of HMMs whose topologies were initialized using the k-variable k-means algorithm with a generality threshold of 1. This system represents a substantial improvement over the best GMM system while using fewer parameters, indicating that it is a better choice for modeling environments.

Confusion matrices for both the seen and unseen conditions are shown in Tables 4.13 and 4.14, F_1 scores per environment in Table 4.15, and top confusions in Tables 4.16 and

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	107	1	8	0	2	2	0	0	0	89.17%
bus	0	119	0	0	0	0	0	1	0	99.17%
galry	0	0	119	0	1	0	0	0	0	99.17%
park	0	0	0	119	0	0	1	0	0	99.17%
plaza	2	0	3	0	115	0	0	0	0	95.83%
restr	0	0	0	0	0	119	0	0	1	99.17%
strt	6	0	0	0	4	1	107	2	0	89.17%
trn	2	3	0	0	1	1	2	111	0	92.50%
platf	5	0	1	2	2	4	1	0	105	87.50%
PRECISION	87.70%	96.75%	90.84%	98.35%	92.00%	93.70%	96.40%	97.37%	99.06%	

Table 4.13: HMM Environmental Confusions, Precision, and Recall: Seen Condition

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	53	0	0	0	0	11	48	0	8	44.17%
bus	4	110	0	0	1	0	0	5	0	91.67%
galry	0	4	116	0	0	0	0	0	0	91.67%
park	0	0	0	120	0	0	0	0	0	100.0%
plaza	0	0	0	0	107	5	7	1	0	89.17%
restr	0	0	2	0	0	118	0	0	0	98.33%
strt	22	0	0	0	18	2	75	0	3	62.50%
trn	0	33	0	0	10	0	0	75	2	62.50%
platf	0	8	80	0	0	0	0	0	32	26.67%
PRECISION	67.09%	70.97%	58.59%	100.0%	78.68%	86.76%	57.69%	92.59%	71.11%	

Table 4.14: HMM Environmental Confusions, Precision, and Recall: Unseen Condition

4.17.

As these confusion summaries show, there were no significant confusion issues on the seen data; with such a small overall error rate, this is to be expected. On the unseen data, however, several confusions are significant. The platform-gallery confusion, which accounted for 7.47% of the total answers and nearly 30% of the total errors, is an interesting one in that one would presume that when an actual train is not in the process of pulling up or leaving, the platform might not in fact be distinguishable from a gallery. The train-bus, street-plaza, airport-restaurant, plaza-street confusions are also understandable in an intuitive way. The remaining significant confusions airport-street and street-airport, are less easy to understand. It is possible that some specific locales in some specific environments

Environment	Total F_1 Score	Seen F_1 Score	Unseen F_1 Score
park	99.37	98.75	100.0
restaurant	94.26	96.35	92.18
bus	88.96	97.94	80.00
plaza	88.73	93.87	83.59
train	84.74	94.87	74.62
gallery	83.15	94.82	71.48
street	76.31	92.64	59.99
airport	70.84	88.42	53.26
train_platform	65.85	92.92	38.79

Table 4.15: HMM F_1 Scores

Rank	Ref	Hyp	Pct	Rank	Ref	Hyp	Pct
1	Airport	Gallery	0.74%	6T	Train	Bus	0.27%
2	Street	Airport	0.55%	6T	Plaza	Gallery	0.27%
3	Platform	Airport	0.46%	8T	*8 tied*		0.18%
4T	Platform	Restaurant	0.37%				
4T	Platform	Plaza	0.37%				

Table 4.16: Top 10 HMM Confusions as Percentage of Total Answers, Seen Condition

Rank	Ref	Hyp	Pct	Rank	Ref	Hyp	Pct
1	Platform	Gallery	7.47%	6	Airport	Restaurant	1.01%
2	Airport	Street	4.44%	7	Train	Plaza	0.92%
3	Train	Bus	3.05%	8T	Platform	Bus	0.74%
4	Street	Airport	2.03%	8T	Airport	Platform	0.74%
5	Street	Plaza	1.67%	10	Plaza	Street	0.65%

Table 4.17: Top 10 HMM Confusions as Percentage of Total Answers, Unseen Condition

do in fact sound like city streets; however, this confusion is less intuitive and likely an indication of a real modeling inadequacy than any true similarity.

The F_1 scores shown above measure the overall difficulty of each class. As with humans, park and restaurant are the easiest classes. Interestingly, train platform is the hardest class, owing mainly to the very poor performance in the unseen dataset. Also interestingly, while airport is hard, the other more ill-defined classes, plaza and gallery, are not among the hardest classes. This is a real difference between human and machine environment recognition systems that will be explored in more detail in Section 4.8 below.

4.7 Experimental Results: Optimal Coding Approach

Here experiments using the optimal coding approach are described, including studies on basic feature selection and model complexity for both ICA and MLP autoencoder trees. After discussing each experiment in turn in Sections 4.7.1 - 4.7.5, a summary of these results and error analysis are presented in Section 4.7.6

Feature	Total Error	Seen Error	Unseen Error
MFCC	70.64%	71.67%	69.62%
MFCC+	79.17%	74.72%	83.61%

Table 4.18: ICA Feature Selection, 10 coding units

4.7.1 ICA Feature Selection

Given the nearly 50% relative difference in performance between MEL-based features and MFCC-based features demonstrated in the experiments on GMM feature selection above, the initial ICA experiment dealt only with the MFCC and MFCC+ feature sets. For this experiment, 10-dimensional ICA solutions were sought for each feature set using the deflation variant of the FastIca algorithm with a $\tanh$ nonlinearity, a stopping criterion of 0.001, and a maximum of 5000 iterations per component. In order to carry out the FastIca procedure, the data were first whitened by applying PCA followed by multiplication by the inverse of the square root of the eigenvalue matrix. After discarding all but the first 10 dimensions of the whitened matrix, the FastIca procedure was carried out to estimate the optimal coding matrix C_χ for each environment. The decoding matrix D_χ was then estimated per Equation 4.1. Performance is shown in Table 4.18.

The main result of note in this experiment is that performance is overall quite poor compared to the GMM/HMM systems above; the MFCC system is only about 3% better absolute than the average human. Oddly, this system actually has a *lower* error rate on unseen data than seen data. This result would seem to indicate that the MFCC-based ICA system is doing a good job of generalizing compared to recognizing seen locales; however, in an absolute sense the error rates are sufficiently bad as to make this feat useless.

Coding Units	Total Error	Seen Error	Heldout Error
2	75.50%	66.20%	84.81%
4	65.92%	58.33%	73.51%
6	68.47%	59.72%	77.22%
8	68.37%	62.22%	74.53%
10	70.64%	71.67%	69.62%
12	79.95%	78.98%	80.92%

Table 4.19: ICA Model Complexity

4.7.2 ICA Model Complexity

The next experiment measured the effects of model complexity on performance. Using the MFCC feature set, the model complexity was varied between 2 and 12 with the same procedure noted above. Results are shown in Table 4.19.

As these results show, optimal performance is achieved with fairly low model complexity; the best system uses only a 4-dimensional coded feature space. Performance degrades with additional coded features, reaching its worst point with 12 coded features. At first glance, this result would appear to be due to overfitting; however, note that performance on the unseen dataset actually has two troughs, at 4 and 10 coded features, instead of a single trough as one might expect if overfitting were a problem. In any case, the poor overall performance of the ICA approach means that this technique is not indicated for environment recognition.

It is likely that the poor performance of the ICA model is due to the fact that the components that are being learned are directions in multidimensional space, rather than spatial clusters in multidimensional space as with HMM/GMM systems. This deficiency can be seen if we consider a case in which two clusters of data points from two different classes

Feature	Total Error	Seen Error	Heldout Error
MFCC	37.03%	28.33%	45.74%
MFCC+	39.90%	31.29%	48.51%

Table 4.20: MLP Feature Selection, 8 Hidden Units

lie along the same vector with respect to the origin, but are separated by a large distance. In this case, the two classes would be linearly separable and can be discriminated with a single Gaussian. However, since both clusters lie at the same angle from the origin, there is no way for a simple ICA model to discriminate between them. This deficiency could in theory be addressed using an ICA mixture model, in which directions from many different origin points are learned. As seen below, however, single MLP autoencoders outperform single ICA autoencoders using this data set and so ICA mixtures are dispreferred in this work in favor of MLP mixtures.

4.7.3 MLP Autoencoder Feature Selection

As noted above, an alternate means of producing an optimal environment-specific code is to train an autoencoding MLP. The initial MLP experiment compared the MFCC feature set to the MFCC+ feature set using 8 hidden units. After whitening the training data, the MLPs were trained using error backpropagation with the mean squared error criterion. Each network weight was initialized at random with a value between -0.05 and 0.05, and the weights had independent, adaptive learning rates (initialized to 0.05) and momentum terms (initialized to 0.045). Each network was trained for 10,000 iterations in batches of 10,000 sample frames. Results are shown in Table 4.20.

These results, while still worse than the GMM/HMM results, are much better than both the human and ICA results. Like the ICA results, the MFCC feature set yielded better results than the augmented MFCC+ feature set, and was used as the feature set for future

Hidden Units	Total Error	Seen Error	Heldout Error
2	37.87%	30.64%	45.09%
4	37.31%	25.18%	49.44%
6	36.57%	26.85%	46.29%
8	37.03%	28.33%	45.74%
10	44.16%	32.50%	55.83%
12	53.47%	47.50%	59.44%

Table 4.21: MLP Hidden Unit Selection

experiments.

4.7.4 MLP Autoencoder Model Size

The second MLP experiment tested the effects of model order. Six additional MLPs per environment were trained with varying numbers of hidden units from 2 to 12. Results are shown in Table 4.21.

The overall optimal system is the 6-unit system. This system is best at neither seen locale recognition nor generalization. However, it appears that the systems which generalize well do dot recognize seen locales (2 units, 8 units), while the converse is also true (4 units). The 6-unit system appears to yield the best tradeoff between these two concerns. It is very interesting, however, that the best generalizer has only two hidden units. The overall poor performance makes it impossible to conclude that two units are all that are needed to capture the essential features of an environment; however, it is clear that *adding* units beyond this level does not help generalization performance using this model. Conversely, it is to be expected that the worst performance is achieved by the 12-unit system. In the autoencoder approach, the closer the number of hidden units is to the signal dimensionality, the closer the coding matrix will be to the unit matrix, which cannot be used to discriminate between

System	Params	Total Error	Seen Error	Heldout Error
6-Unit MLP	1,512	36.57%	26.85%	46.29%
6-Gaussian GMM	1,566	31.94%	24.62%	39.25%
HMM	28,756	15.18%	5.09%	25.27%

Table 4.22: MLP vs. GMM with Similar Parameter Spaces

different classes of phenomena at all.

4.7.5 MLP Autoencoder Trees

One issue with MLP autoencoders is that the number of parameters is limited by the dimensionality of the data. The optimal system above, a 6-unit MLP, has only 168 parameters per class for a total of 1,512. By contrast, the optimal HMM system, achieving an error rate of 15.18%, has 3,195 parameters per class for a total of 28,756 — nearly 20 times as many parameters as the best MLP. More importantly, MLP autoencoders suffer from the same flaw as ICA models — each hidden unit is modeling a direction rather than a spatial cluster. What is not clear is whether or not allowing the MLP to select more directions would result in a decrease in the performance gap between HMM/GMM systems and MLPs. To help answer this question, a GMM system was trained with only 6 Gaussians per class. Results of this experiment are shown in Table 4.22.

This result indicates that the difference between MLP and HMM performance is due largely, though not completely, to the larger parameter space that can be employed with HMMs. The 6-Gaussian GMM system still performs better than the 6-unit MLP, but the difference is not enormous. It is possible that by allowing the MLP to model more directions, performance could approach HMM performance.

In order to test this assertion, a mixture of MLPs was constructed for each class as

System	Params	Total Error	Seen Error	Unseen Error
Single MLP	1,512	34.7%	26.4%	43.1%
Depth-2 MLP Tree	3,024	37.6%	26.4%	48.8%
Depth-3 MLP Tree	6,048	36.6%	25.5%	47.8%
Depth-4 MLP Tree	12,096	37.1%	23.8%	50.4%
Depth-5 MLP Tree	24,192	32.6%	21.2%	44.1%
Depth-6 MLP Tree	48,384	32.1%	21.0%	43.1%
Depth-7 MLP Tree	96,768	30.1%	18.9%	41.4%

Table 4.23: MLP Tree Depth

described in Section 4.4.2. This approach allows the training data to become segmented into pools of similar kind, or equivalently, to create multiple origins in feature space from whence to learn directions. The mixtures were arranged as binary trees in order to more easily compare results between levels. Binary trees of autoencoding MLPs were trained down to 7 levels; these models had 64 MLPs at the leaves. Each level was evaluated; results of this experiment are shown in Table 4.23.

After degrading in levels 2, 3, and 4, performance improves in subsequent levels and reaches a maximum at depth 7, with an error rate of 30.1%. Both seen locale recognition and generalization are optimized at this level. As expected, seen locale recognition improves monotonically with increasing number of parameters. However, generalization degrades before finally improving, suggesting that data fragmentation helps generalization as well as locale recognition by allowing some networks to focus on very fine details of the sound field. One issue with the level 7 networks, however, is that data fragmentation may lead to poor modeling. Specifically, with 2.7 million training frames per environment, there are only on average 250 training examples per parameter. Compared to the optimal HMM system, which has 845 training examples per parameter, this is a small figure. It is possible

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	86	2	14	0	4	13	1	0	0	71.67
bus	0	119	1	0	0	0	0	0	0	99.17
galry	1	0	115	0	0	3	0	1	0	95.83
park	0	0	0	116	0	0	4	0	0	96.67
plaza	4	3	16	1	83	3	10	0	0	69.17
restr	4	0	0	0	0	114	2	0	0	95.00
strt	5	0	0	3	6	11	90	4	1	75.00
trn	5	15	0	7	0	10	2	81	0	67.50
trnp	9	3	9	8	3	8	5	3	72	60.00
PRECISION	75.44	83.80	74.19	85.93	86.46	70.37	78.95	91.01	98.63	

Table 4.24: Depth-7 MLP Tree Environmental Confusions, Precision, and Recall: Seen Condition

to extend the MLP tree to another level and employ 128 leaf MLPs per class, but further data fragmentation might lead to some very poorly-trained networks; further, the performance trend does not seem to indicate that the added directions would allow the MLP tree to achieve performance levels comparable to the best HMM systems. The main conclusion that can be drawn from this experiment is that, in fact, the optimal coding approach is inferior to a more standard GMM/HMM approach for the environment recognition task.

4.7.6 Optimal Coding Result Summary

Confusion matrices for both seen and unseen conditions are shown in Tables 4.24 and

4.25, F_1 scores per environment in Table 4.26, and top confusions in Tables 4.27 and 4.28.

As these confusion summaries show, the seen condition had some significant confusions. Many of these were understandable, e.g., train / bus, airport / gallery, airport / restaurant, and plaza / street. The top confusion, plaza / gallery, is somewhat disappointing, as this confusion is an indoor / outdoor confusion. On the unseen data, there were 3 confusions which were particularly damaging to system performance. The plaza / park, platform / gallery, and airport / street confusions together accounted for over 40% of the

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	18	0	0	2	0	31	69	0	0	15.00
bus	0	119	0	0	0	0	0	1	0	99.17
galry	0	8	109	0	0	0	0	0	3	90.83
park	0	0	0	115	0	0	3	0	2	95.83
plaza	1	0	0	107	2	0	10	0	0	1.67
restr	0	0	1	0	1	117	1	0	0	97.50
strt	10	0	0	0	6	13	89	0	2	74.17
trn	0	13	1	0	20	0	9	54	23	45.00
trnp	0	22	88	0	0	0	0	0	10	8.33
PRECISION	62.07	73.46	54.77	51.34	6.90	72.67	49.17	98.18	25.00	

Table 4.25: Depth-7 MLP Tree Environmental Confusions, Precision, and Recall: Unseen Condition

Environment	Total F_1 Score	Seen F_1 Score	Unseen F_1 Score
bus	87.90	90.83	84.99
restaurant	82.06	80.85	83.27
park	78.92	90.98	66.86
gallery	75.97	83.63	68.33
train	69.60	77.51	61.71
street	68.02	76.92	59.13
airport	48.82	73.50	24.16
train_platform	43.55	74.61	12.49
plaza	40.76	78.85	2.68

Table 4.26: MLP Tree F_1 Scores

Rank	Ref	Hyp	Pct	Rank	Ref	Hyp	Pct
1	Plaza	Gallery	1.48%	6T	Plaza	Street	0.92%
2	Train	Bus	1.38%	6T	Train	Restaurant	0.92%
3	Airport	Gallery	1.29%	8T	Platform	Airport	0.83%
4	Airport	Restaurant	1.20%	8T	Platform	Gallery	0.83%
5	Street	Restaurant	1.01%	10T	*2 Tied*		0.74%

Table 4.27: Top 10 MLP Tree Confusions as Percentage of Total Answers, Seen Condition

Rank	Ref	Hyp	Pct	Rank	Ref	Hyp	Pct
1	Plaza	Park	9.90%	6	Platform	Bus	2.03%
2	Platform	Gallery	8.14%	7	Train	Plaza	1.85%
3	Airport	Street	6.33%	8T	Train	Bus	1.20%
4	Airport	Restaurant	2.87%	8T	Street	Restaurant	1.20%
5	Train	Platform	2.12%	10T	*2 Tied*		0.92%

Table 4.28: Top 10 MLP Tree Confusions as Percentage of Total Answers, Unseen Condition

total errors made in the unseen condition. The platform / gallery and airport / street confusions were problematic for the HMM system as well, but plaza / park was not. In fact, the F_1 score for the park class was 100 in the unseen condition for HMMs, but only 67 in the MLP system. This modeling failure is a large source of the difference between HMM and MLP performance. Many of the other top MLP confusions are also in the set of top HMM confusions, meaning that there are only a few classes where distinctly different types of confusions are being made. Were the error types vastly different, a hybrid HMM / MLP system could be considered; given these results, together with the overall performance gap between the two system types, it is unlikely that such an approach would yield substantial improvements.

4.8 Comparison of Human and Automatic Performance

Results shown above indicate that on average, machine performance far exceeds human performance on the acoustic environment recognition task. However, it could be the case that the human subset of the evaluation data was particularly hard or easy, which would skew the results. In order to avoid misinterpretation, the human subset was used to test both the HMM and Depth-7 MLP tree systems. These results are shown in Table 4.29, together with complete evaluation set results for comparison.

As these results show, the human subset was significantly harder for the machine systems than the remainder of the evaluation set. Both the HMM and MLP lost 14% absolute overall. While the HMM lost 17% on seen locale recognition and 11% on generalization, the MLP lost 11% on seen locale recognition and 17% on generalization. Confusion matrices and F_1 scores for seen and unseen data with both HMM and MLP systems are shown in Tables 4.30 - 4.34.

The confusions and F_1 scores on the human subset are in some ways revealing; however, the reader is cautioned that since this subset only contained 12 example segments per

System	Total Error	Seen Error	Unseen Error
Humans	73.7%	74.4%	72.9%
HMM	29.6%	22.2%	37.0%
Depth-7 MLP Tree	44.4%	29.6%	59.2%
HMM - All Eval	15.2%	5.1%	25.3%
MLP - All Eval	30.1%	18.9%	41.4%

Table 4.29: Error Rates on Human-Evaluated Subset

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	3	0	2	0	0	1	0	0	0	50.00
bus	0	6	0	0	0	0	0	0	0	100.00
galry	0	0	6	0	0	0	0	0	0	100.00
park	1	0	0	1	1	0	2	0	1	16.67
plaza	1	0	0	0	5	0	0	0	0	83.33
restr	0	0	0	0	0	6	0	0	0	100.00
strt	1	0	0	0	0	1	4	0	0	66.67
trn	0	0	0	0	0	0	0	6	0	100.00
trnp	0	0	0	1	0	0	0	0	5	83.33
PRECISION	50.00	100.00	75.00	50.00	83.33	75.00	66.67	100.00	83.33	

Table 4.30: Environmental Confusions, Precision, and Recall: HMM, Human Seen Subset

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	3	0	0	0	0	0	3	0	0	50.00
bus	0	4	0	0	2	0	0	0	0	66.67
galry	0	0	4	0	0	0	0	2	0	66.67
park	0	0	0	6	0	0	0	0	0	100.00
plaza	0	0	0	0	5	0	1	0	0	83.33
restr	0	0	0	0	0	6	0	0	0	100.00
strt	2	0	0	0	0	1	1	2	0	16.67
trn	0	3	0	0	1	0	0	2	0	33.33
trnp	1	1	1	0	0	0	0	0	3	50.00
PRECISION	50.00	50.00	80.00	100.00	62.50	85.71	20.00	33.33	100.00	

Table 4.31: Environmental Confusions, Precision, and Recall: HMM, Human Unseen Subset

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	3	0	2	0	0	1	0	0	0	50.00
bus	0	6	0	0	0	0	0	0	0	100.00
galry	0	0	6	0	0	0	0	0	0	100.00
park	1	0	0	4	0	0	1	0	0	66.67
plaza	0	0	1	0	5	0	0	0	0	83.33
restr	0	0	0	0	0	6	0	0	0	100.00
strt	2	0	0	0	0	1	3	0	0	50.00
trn	1	1	0	0	0	1	0	3	0	50.00
trnp	0	0	2	2	0	0	0	0	2	33.33
PRECISION	42.86	85.71	54.55	66.67	100.00	66.67	75.00	100.00	100.00	

Table 4.32: Environmental Confusions, Precision, and Recall: D-6 MLP Tree, Human Seen Subset

Reference	Hypothesis									
	airpt	bus	galry	park	plaza	restr	strt	trn	platf	RECALL
airpt	0	0	0	0	0	1	5	0	0	0.00
bus	0	5	1	0	0	0	0	0	0	83.33
galry	0	0	4	0	1	0	0	1	0	66.67
park	0	0	0	5	0	0	0	0	1	83.33
plaza	0	0	3	3	0	0	0	0	0	0.00
restr	0	0	0	0	0	5	1	0	0	83.33
strt	2	0	0	0	0	2	2	0	0	33.33
trn	0	4	0	0	1	0	0	1	0	16.67
trnp	2	0	3	0	0	1	0	0	0	0.00
PRECISION	0.00	55.56	36.36	62.50	0.00	55.56	25.00	50.00	0.00	

Table 4.33: Environmental Confusions, Precision, and Recall: D-6 MLP Tree, Human Unseen Subset

Environment	Human	HMM		Depth-6 MLP Tree	
	Total F_1	Seen F_1	Unseen F_1	Seen F_1	Unseen F_1
airport	12.06	50.00	50.00	46.15	0.00
bus	23.88	100.0	57.14	92.30	66.67
gallery	14.01	85.71	72.73	70.59	47.05
park	62.54	25.00	100.0	66.67	71.42
plaza	15.32	83.33	71.42	90.90	0.00
restaurant	32.23	85.71	92.30	80.00	66.67
street	19.17	66.67	18.18	60.00	28.57
train	29.95	100.0	33.33	66.67	25.00
platform	19.29	83.33	66.67	66.67	0.00

Table 4.34: F_1 Scores on Human Subset

environment (6 seen, 6 unseen), the specific figures are not as reliable as figures drawn from the entire test set. Conversely, the F_1 scores shown in Table 4.36 compare human results on the human subset to machine results on the entire test set, which is not a direct comparison. With this warning in mind, the environments are shown ranked by difficulty on the human subset in Table 4.35.

While some classes are similarly difficult for humans and machines, there are significant ordinal differences in environmental difficulty ratings. Though overall performance is much worse for humans, relative difficulty is worth exploring. There are large differences in relative difficulty for several environment classes between human performance and HMM performance. Galley and plaza are more difficult for humans than for HMMs, while park, train, and street are more difficult for HMMs. As noted above, however, these machine results are not necessarily representative of true performance, as they are based (like the human results) on only 5% of the test data. Hence, it is somewhat difficult to draw

Difficulty	Human		HMM		MLP Tree	
Rank	Env	F_1	Env	F_1	Env	F_1
1	Airp	12.06	Strt	42.42	Airp	23.07
2	Gall	14.01	Airp	50.00	Plat	33.33
3	Plaz	15.32	Park	62.50	Strt	44.28
4	Strt	19.17	Trn	66.67	Plaz	45.45
5	Plat	19.29	Plat	75.00	Trn	45.83
6	Bus	23.88	Plaz	77.37	Gall	52.82
7	Trn	29.95	Bus	78.56	Park	69.04
8	Rest	32.23	Gall	79.21	Rest	73.33
9	Park	62.54	Rest	89.00	Bus	79.48

Table 4.35: Environment Difficulty Ranked By F_1 Scores on Human Subset

conclusions based on these results. For a comparison based on all test data, see Tables 4.36 and 4.37.

These full comparisons still show systematic differences between human and machine performance. Compared to HMMs, humans have less relative difficulty with train and platform, and more relative difficulty with gallery and plaza. Conversely, humans, like HMMs, perform best on the park and restaurant classes. As noted above, one might expect humans to perform relatively poorly on such ill-defined classes as plaza and gallery. Further, one might expect HMMs to perform relatively poorly on a class like train platform, simply because when there is no train present, a train platform sounds very much like a generic gallery. Humans might not be tricked as easily by this condition, especially if certain kinds of speech are present in the signal that one might associate with a train station: announcements of train arrival, for instance, are easily recognized by humans, while to a very simple HMM like the one used here, such announcements might be indistinguishable from an air-

Environment	Human	HMM		Depth-6 MLP Tree	
	Total F_1	Seen F_1	Unseen F_1	Seen F_1	Unseen F_1
airport	12.06	88.42	53.26	73.50	24.16
bus	23.88	97.94	80.00	90.83	84.99
gallery	14.01	94.82	71.48	83.63	68.33
park	62.54	98.75	100.0	90.98	66.86
plaza	15.32	93.87	83.59	78.85	2.68
restaurant	32.23	96.35	92.18	80.85	83.27
street	19.17	92.64	59.99	76.92	59.13
train	29.95	94.87	74.62	77.51	61.71
platform	19.29	92.92	38.79	74.61	12.49

Table 4.36: F_1 Scores on All Data

Difficulty	Human		HMM		MLP Tree	
Rank	Env	F_1	Env	F_1	Env	F_1
1	Airp	12.06	Plat	65.85	Plaz	40.76
2	Gall	14.01	Airp	70.84	Plat	43.55
3	Plaz	15.32	Strt	76.31	Airp	48.82
4	Strt	19.17	Gall	83.15	Strt	68.02
5	Plat	19.29	Trn	84.74	Trn	69.60
6	Bus	23.88	Plaz	88.73	Gall	75.97
7	Trn	29.95	Bus	88.96	Park	78.92
8	Rest	32.23	Rest	94.26	Rest	82.06
9	Park	62.54	Park	99.37	Bus	87.90

Table 4.37: Environment Difficulty Ranked By F_1 Scores on All Data

port announcement. Relative difficulty aside, this result underscores exactly how much more difficult the acoustic environment recognition task is for humans than for machines — the human subjects' F_1 score on their best class, park, is still lower than the HMM's score on it's worst class, train platform. As most humans are primarily conditioned to use the audio channel for full-field danger detection and speech communication, they are ill-equipped to attempt to distinguish between the environmental types studied here. Different results might obtain were a different class of human subject tested; those humans who are either visually impaired or specially trained might perform better. In the small pilot study described in [63], exposing the human subjects to the same training data that the machines used decreased overall error rate from 21.2% to 11.8%, though this was still much worse than the best machine error rate of 1.7% in that study.

4.9 Adaptation to Unseen Locales

The performance gap between segments from locales seen in training and those unseen in training — 5.1% for the former versus 25.3% for the latter in the best system — suggests that while there is some capacity for generalization, in general the penalty for straying outside of known locales is stiff. Further, as shown above, the range of F_1 scores is much narrower for seen locales than for unseen locales. For the best HMM system, for instance, the F_1 range for the seen condition is approximately 88 - 98, while the F_1 range for the unseen condition is 38 - 100. Since only one recording was set aside per environment for the unseen condition, it is possible that these figures may not represent the average case unseen condition performance. To help determine whether or not this was the case, an additional round-robin experiment was carried out using only the GMM system with model order ranging from 12 to 125 Gaussians. For this experiment, each recording was held out in turn for testing while all the remaining data were used for training. Recall that the best GMM system had an overall error rate of 17% using the original data assignment

into training, seen evaluation, and unseen evaluation, with 8% error on seen locales and 26% on unseen locales.

The best result for this experiment was obtained using 100 Gaussians; the error rate was 50.8%. In addition, the range of results was exceptionally narrow; the worst system, using 25 Gaussians, had an error rate of 52.8%. This figure is significantly worse than the 25% error rate reported for the unseen condition of the 2006 CLEAR evaluation. While it is still much better than chance, and much better than human performance, this level of performance does not suggest readiness for application. On further inspection, though, it appeared that the results were essentially bimodal. As shown in Figure 4.2 and Figure 4.3, there were a large number of recordings with error rates at or near zero, and another, smaller but still large contingent of recordings with error rates near one.

There are two ways to further examine these figures: first, by environment; second, by locale or country. Table 4.38 shows, for each environment, the miss rates both in the round-robin condition and in the unseen portion of the original experiment. The fact that these systems used different classifiers is less important than the fact that the miss rates track reasonably closely between the two conditions, at least in relative terms. That is, the ordering of environments from hard to easy is quite similar. The results shown in this table indicate that, on average, previous assessments regarding the relative difficulty of each environment type were reasonably accurate. Results broken down by country are found in Table 4.39. Note that for some environments, cases in which there is only one recording from a country have high error rates, and for others, the opposite is true.

Given the miss rates in the round-robin test condition, it makes sense to ask next how much data is required to improve performance to acceptable levels. In order to answer this question, a new round-robin test was carried out in which, after initial training and testing, data from the test recording was added to the training data a minute at a time. This data was used to update the parameters of the acoustic model for the test environment. After each minute of adaptation data was added to the model, the remaining data in the

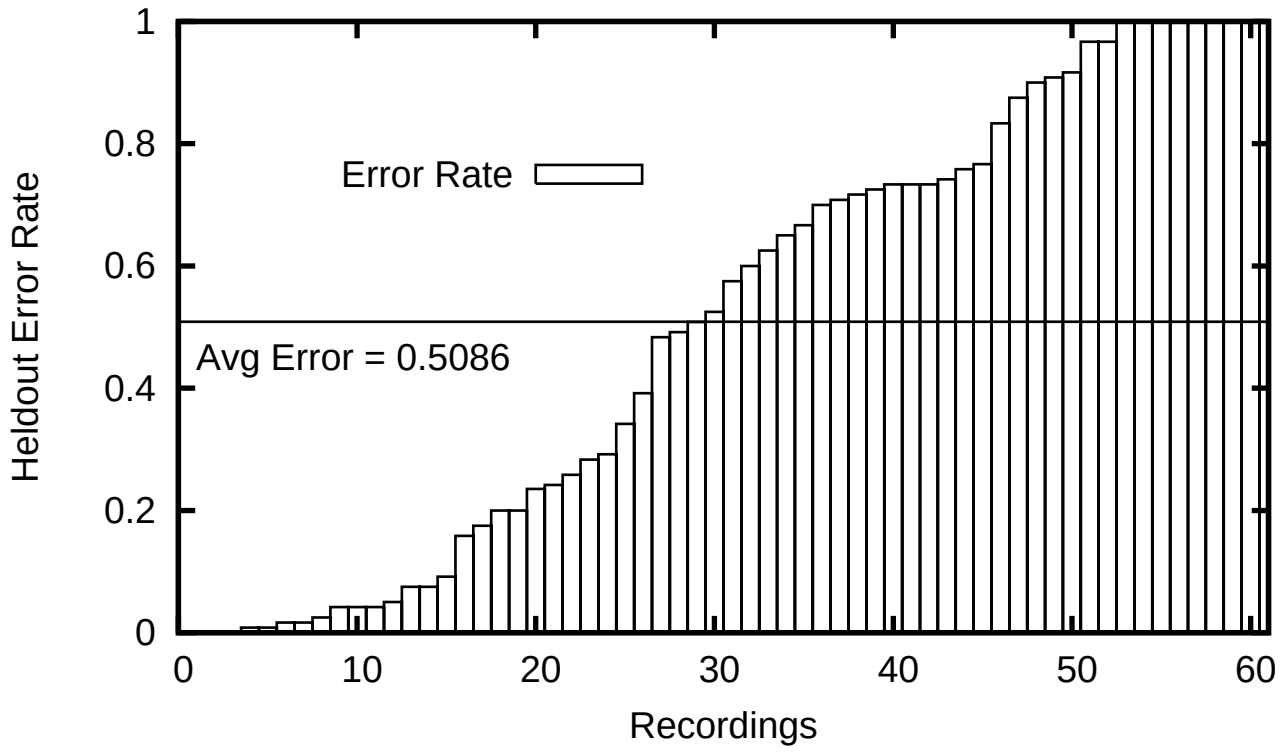

Figure 4.2: Round-Robin Error Rates per Recording

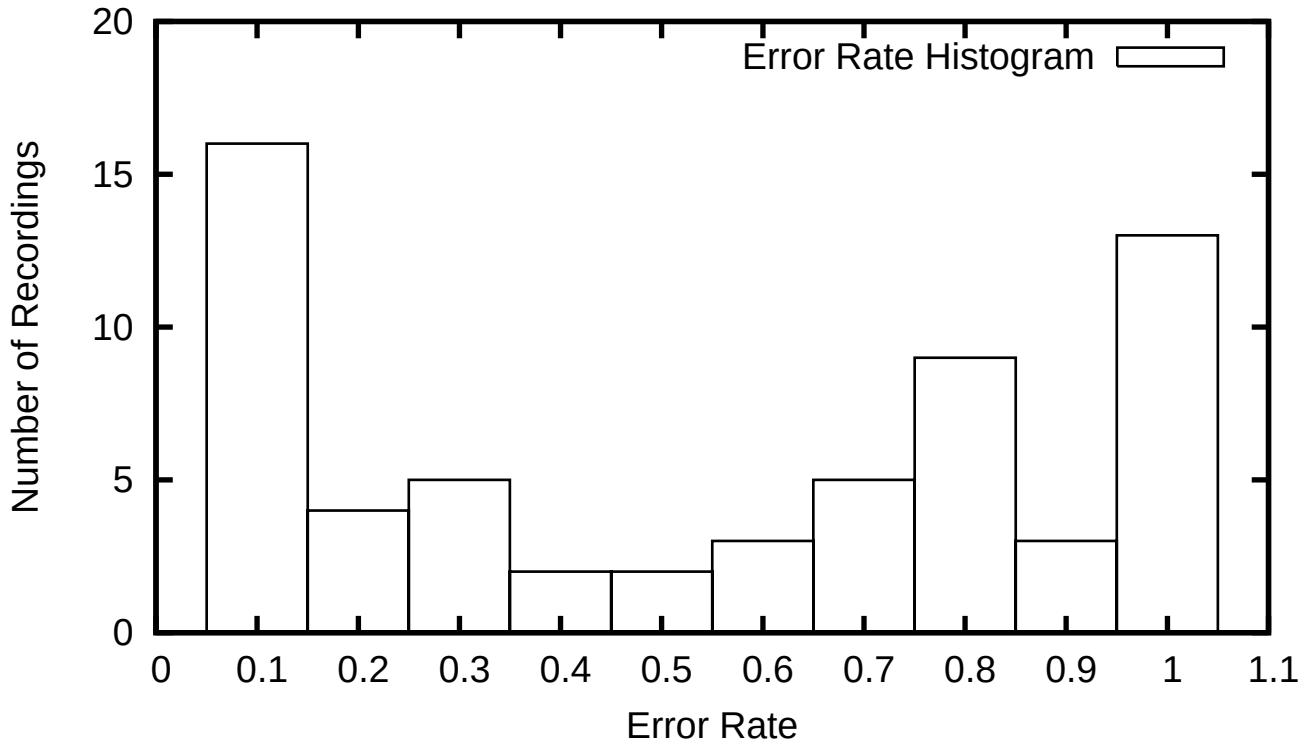

Figure 4.3: Round-Robin Error Rate Histogram

Environment	Round-Robin		Original Unseen	
	GMM Miss Rate	Rank	HMM Miss Rate	Rank
Airport	54.16	5	55.83	2
Bus	20.11	9	8.33	6
Gallery	17.97	8	8.33	6
Park	39.78	6	0.00	9
Plaza	68.69	3	10.82	5
Restaurant	37.38	7	1.67	8
Street	62.22	4	37.50	3
Train	76.25	2	37.50	3
Train Platform	80.71	1	73.33	1

Table 4.38: Per-Environment Round-Robin Miss Rates and Difficulty Rankings

Environment	Country								
	FRA	GER	GRE	ITA	JAP	SAF	KOR	UK	USA
Airport	- 0	46.9 3	28.3 1	- 0	- 0	57.5 1	100 1	- 0	52.5 1
Bus	- 0	20.5 4	- 0	9.1 1	- 0	- 0	29.2 1	- 0	- 0
Gallery	- 0	8.3 4	- 0	0.4 2	100 1	- 0	- 0	- 0	- 0
Park	- 0	45.7 7	- 0	- 0	- 0	4.1 1	- 0	- 0	- 0
Plaza	5.0 1	- 0	- 0	37.1 2	82.4 2	73.3 1	- 0	- 0	- 0
Restaurant	0.8 1	67.9 3	- 0	- 0	13.7 2	- 0	- 0	24.1 1	- 0
Street	66.7 1	65.4 2	- 0	- 0	48.3 1	65.0 1	- 0	62.5 1	- 0
Train	60.0 1	73.7 4	- 0	- 0	73.7 2	- 0	- 0	- 0	- 0
Train Platform	100 1	72.9 5	- 0	- 0	100 1	- 0	- 0	- 0	- 0

Table 4.39: Per-Environment and Per-Country Round-Robin Miss Rates and Counts

Adapt Minutes	Number of Gaussians					
	12	25	50	75	100	125
0	52.07	52.81	51.85	52.12	50.86	52.25
1	46.89	44.41	43.49	41.67	41.61	41.05
2	43.43	40.05	37.07	35.79	35.39	35.26
3	40.17	36.85	32.79	31.74	30.01	31.08
4	37.79	32.66	30.56	28.73	27.29	27.95
5	35.37	30.42	28.28	26.59	24.79	25.37
6	34.47	28.94	26.59	25.46	24.39	25.75
7	33.00	27.26	24.96	23.22	22.71	23.90
8	30.67	25.01	22.71	21.59	20.72	22.58
9	31.40	25.37	22.82	22.24	22.13	22.24

Table 4.40: Adaptation to Unseen Locales, GMM System

test recording was evaluated. This experiment was carried out for GMMs ranging in size from 12 Gaussians to 125 Gaussians. Results are shown in Table 4.40. After five minutes of adaptation, the average error rate is down from 50.86% to 24.79%; indicating that in practice not much data is needed to bring performance to an acceptable level. After eight minutes, error rates are down to 20.72%, which is certainly an operationally useful figure — especially considering that this approach uses *only* acoustic information and can be coupled with user-specific priors and transition models to improve performance.

4.10 Chapter Summary

This chapter presented two different methods for recognizing environmental type from audio; one based on conventional GMM/HMM modeling, the other on an optimal coding

/ redundancy reduction modeling approach. The two main results given here are first, that the GMM/HMM approach appears to be superior to the optimal coding approach, and second, that the machine approaches appear to be superior to human performance by a wide margin. The best system trained achieved an error rate of 5.1% on seen locales, and 25.2% on unseen locales as measured using the 2006 NIST CLEAR evaluation data. This performance handily eclipsed the average human performance of 73.7% error. While it is likely the case that specially-trained humans might improve on this result substantially, it is an open question as to whether they might approach 75% accuracy. Even if they can, there are certain applications, e.g., signals analysis, which still might be better handled by machines, which do not suffer from common problems seen with human annotators, like bias or fatigue.

The considerable performance gap between machine performance on data from locales seen in training and locales not seen in training indicates that, as one might expect, generalization performance lags behind locale recognition. Round-robin experiments confirmed that in fact the generalization gap is much larger than is apparent from the CLEAR evaluation data. Performance on unseen data using the round-robin evaluation was still much better than chance and much better than human performance, and thus one can claim that some form of generalization is still taking place. However, the bimodality of results in this experiment leads to a slightly different conclusion: namely, that some environments are easier to extract general features from than others, especially when the data are drawn from different countries and continents. This claim is intuitively plausible; for instance, airports should be and in fact are hard to recognize in this framework. Fortunately, supervised adaptation results demonstrate that one does not need a tremendous amount of adaptation data to bring performance to acceptable levels. In practice, one might imagine that most of an individual's time is spent in familiar rather than new locales, and so, from an application point of view, these deficiencies need not be crippling.

Chapter 5

Auditory Scene Learning for Multiresolution Context Awareness

5.1 Chapter Overview

This chapter describes theoretical and practical work on auditory scene learning and identification for multiresolution indexing and awareness. It begins by motivating multiresolution indexing, and describing how machine listening can help to provide it. After discussing prior work in this area, including a brief discussion of low-level acoustic event detection and recognition, the presentation moves on to methods for modeling, learning, and identifying high-level structures, or acoustic scenes. Experiments are presented which demonstrate the ability to learn these high-level scenes from unlabeled data, and to use them for a specific context awareness goal motivated by the requirements of the CHIL Connector service: the ability of smart mobile telephones to determine from audio information whether or not the user is interruptible.

5.2 Multiresolution Indexing and Awareness

Multiresolution awareness is something that humans do effortlessly. We recognize short-term events in the sensory stream, such as passing a person on the street, hearing a knock on the door, or listening to another person speak. Yet this is not the only level of awareness we have; we recognize that these events are part of larger structures which we also easily recognize. Passing people on the street is park of going to work, the knock at the door is part of a typical workday and a sign that a meeting is about to start, the words coming from another person's mouth are part of a conversation that is itself part of a series of conversations about some topic of shared interest. Further, this high-level awareness, in addition to helping us navigate through the day, also helps to guide low-level sensory recognition systems; recent research has indicated [find citation] that the flow of sensory information is far from a one-way street. In addition to the expected information flow from the sensory periphery to processing areas of the brain, the brain also sends a tremendous amount of information *back* to the sensory periphery. This means that to some presumably large degree, our high-level knowledge and memory are guiding us in our most basic perceptual tasks. Just as high-level knowledge is exploited by biological systems, so too can it be exploited by computational systems by allowing these systems to make better predictions about which short-term events are likely to occur.

This observation has tremendous implications for perceptual computing in general and for context-aware smart spaces and devices in particular. Given that it is important for these systems to appropriately react to changes in context, that these context cues are often short-term events, and that these short-term events are often predictable from high-level knowledge, it is crucial for these systems to attain multiple levels of awareness, from short-term and concrete to long-term and abstract. For multimedia indexing applications, the same requirement holds; in order to effectively index a large multimedia collection for easy human browsing, one cannot simply divide up the stream into an unending sequence

of low-level events. High-level structures, presented at a much slower rate are a necessity for usability.

As with other sensory applications, and for reasons stated earlier in this thesis, the audio signal is a rich source of information which can be exploited to attain multiresolution context awareness. Just as, for instance, many *environments* have characteristic sound fields, many *activities* have characteristic, acoustically detectable events and environmental conditions. Concentrated work in an office environment, for instance, often sounds quite different from informal discussions in the same environment. The former is typically dominated by the sounds of keyboard and mouse clicking, while the latter is typically dominated by speech and other non-speech human noises. While an environment recognition system might correctly classify both of these situations as office scenarios, a multiresolution activity detection system might be able to distinguish them based on short-term and medium-term acoustic cues and use this information to provide the appropriate services. One might, for instance, prefer different office lighting levels for programming work and conversation, or typically make coffee or tea for conversations, or like to listen to music while programming. To the extent that these activities can be recognized, and that sufficient preference-learning algorithms are available, these preferences can potentially be met by a smart space acting autonomously. Given the immense promise of such smart applications, multiresolution auditory scene learning is an extremely attractive area to address.

5.3 Prior Work

Much of the prior work on context awareness from sensory data focuses on either low-level audio context cues or multimodal inputs. Examples of low-level audio context systems include work on speech activity detection or acoustic event classification and detection. Examples of multiresolution, multimodal context systems include audiovisual personal diary or broadcast television segmenters. More specific examples of these kinds of systems

are given below.

5.3.1 Low-Level Context Cues

There is a fairly large body of work on classifying non-speech acoustic events in controlled settings; e.g., for content-based information retrieval. In this application, recordings of isolated sounds are classified or clustered into one of several known classes.

The systems presented by Slaney in [90] and [89] use MFCCs and GMMs to build hierarchical clusters of sounds for retrieval. In this work, a GMM Θ_i, representing a cluster C_i, is trained for each recording. Clusters are then merged according to the criterion

$$\Phi(C_1, C_2) = \mathcal{L}(C_1|\Theta_2) + \mathcal{L}(C_2|\Theta_1), \tag{5.1}$$

where $L(\cdot|\cdot)$ is the likelihood of the data given the model. Clustering proceeds until some stopping criterion is reached (e.g., a specific number of clusters, all clusters merged, high Φ, etc.). The model can then be used to classify future examples.

Reyes-Gomez and Ellis presented in [83] an HMM-based approach to modeling general audio events in the MuscleFish database (introduced by Wold et. al. in [112] for classification. This work dealt with the problem of selecting an HMM structure for a generic audio class. This is actually a very important area to address, as, unlike ASR, it is not clear *a priori* what kind of model structure is appropriate for any given sound class. Two basic approaches were explored: the k-variable k-means algorithm, given in Figure 4.1, and a GMM-EM approach in which ever more complex models were iteratively grown and trained until some stopping criterion was reached. Three criteria were tested in the GMM-EM framework. The low entropy criterion halts when any self-loop probability falls below some threshold value. The low state occupancy criterion halts when the number of frames assigned to any state falls below some threshold value. The BIC method selects the single model which maximizes the BIC. Optimal results were obtained using the low entropy

criterion, though combining scores from multiple different k-variable k-means models resulted in similar performance.

Feature selection in general audio modeling is also an open question. Many researchers have had success with MFCCs, but in some domains other feature sets yield improvements. One example is work done on general sound recognition in the kitchen environment, due to Kraft et. al. [56], [55]. In this work, temporal ICA is used to improve performance over single-frame ICA features and raw MFCCs. Temporal ICA involves feature stacking as described in Section 2.5.1, following which the feature space is reduced in dimensionality by using PCA. The new features are then whitened and used to compute an ICA solution. As shown in [55], ICA tended to learn multi-frame structures like onsets and offsets which were useful for kitchen sound classification.

Another issue raised by Reyes-Gomez and Ellis was the role of discriminative training. Noting that while approaches based on purely discriminative methods like support vector machines (SVMs), for example, Guo and Li's work ([39], [40]), often provided performance superior to generative models on classification tasks, these approaches were ill-suited for online detection of acoustic events in real environments. The reason for this poor fit is that SVMs rely on pre-segmented data, which is not available in online applications. Temko et. al. reported on isolated sound classification in *real* environments in the context of the CHIL project in [66], [101] and [100]. Their system used a sequence of two-class SVMs, each with an optimally-derived feature set, to distinguish between acoustic events found in seminar and meeting settings. These systems, though relying on presegmented data, were important in that they were used to successfully classify acoustic events produced by spontaneous human activity in real environments. Further work, described in [99], demonstrated the ability to detect acoustic events in online streams using new CHIL data. Unfortunately, neither Temko's SVM system nor the HMM systems produced by Zieger and Malkin were able to perform much better than 100% error due to an astonishing number of insertion errors. This problem was due to the acoustic similarity be-

tween sounds which had task-relevant semantics and those which did not, and also because there was a large amount of speech which tended to confuse the detection systems. For this reason, future work in this domain will likely be integrated with speaker activity detection (SAD) systems.

Finally, as shown in work by Chen et. al. ([12], [13]), even essentially anonymous acoustic event detection — that is, specifying that an event is occurring without identifying it — can be useful as a preprocessing step for more complex video-based activity detection systems. In their work, a very simple power-based acoustic event detector was used to pre-select regions of interest in nursing home videos; these regions of interest were then evaluated for human activity using visual features. Using anonymous acoustic event detection as a preprocessing step in this application resulted in many fewer regions being processed with more expensive visual algorithms.

5.3.2 Scene Learning and Recognition

There is a large body of work on finding meaningful temporal patterns in all types of data. Phrase finding, word clustering, variable-length language modeling, text topic detection, ASR punctuation insertion, and word unit finding in texts from unpunctuated languages can all be viewed as structure or scene learning problems. In multimedia data, the emphasis has been on finding multiresolution structures in, for example, sports video, personal diary recordings, and similar large, relatively unstructured databases. Much of this research has used multimodal inputs including audio, video, and even other more esoteric information sources. Several models and methods have been proposed for this task; two of the most popular approaches involve the use of HMM variants. These variants are now discussed in turn.

The Layered HMM

A classic example of this type of scene learning is found in the many works by Clarkson ([17], [20], [18], [19], [16]) and collaborators. The first two of these works addressed extracting context information in a mobile setting from wearable audio sensors only; specifically, he used MFCCs sampled at a slow frame rate (5 Hz). Clarkson met with limited success with this approach, as many of the locales that he was interested in modeling were very similar acoustically. In later work, he thus added some simple video and acceleration information, fused at the feature level. These additional features improved performance considerably.

In early work, Clarkson trained a single HMM using the segmental k-means algorithm, found in [81]. In practice, this approach required that the number of states be selected *a priori*, though in principle the HMM topology could be learned directly from data using, e.g., one of the approaches found in [83]. In later work, Clarkson made use of multiresolution structures to represent and learn high-level activities. He used the layered HMM (LHMM) to model these structures.

The LHMM consists of a series of HMMs interoperating in the following way. The bottom layer HMM produces discrete symbols or real-valued vectors in the usual way; inference involves estimating the most likely state sequence given observable data. This state sequence, represented either as a real-valued vector of time-dependent state posteriors or discrete state indices, is then used as the observable data by a higher-level HMM, and so on. These models are trained with the standard EM procedures; typically either forward-backward training or Viterbi training. Inference is carried out in a bottom-up fashion; the Viterbi procedure is first used on the bottom layer in order to produce a state sequence which is used for inference on the next higher level.

The main advantages of the LHMM are simplicity and adaptability. The model is trained with standard procedures, and assuming that high-level statistics are similar across

$$O_t^0 \leftarrow S_t^0, \tag{5.2}$$

$$O_t^i \leftarrow S_t^{i-1}, S_t^i, \tag{5.3}$$

$$S_t^i \leftarrow S_{t-1}^0. \tag{5.4}$$

$$\tag{5.5}$$

Figure 5.1: Layered HMM Dependencies

locales, a model trained in one locale is easily adapted to another locale by simply re-training the bottom-level acoustic parameters. There are three main disadvantages with the LHMM. First, the Viterbi procedure must be executed multiple times in sequence, meaning that decoding time for a T-length sequence using an M-level LHMM with N states per level on average $O(MN^2T)$ time. Second, there is no principled way to ensure that high-level state changes are less frequent than low-level state changes. Even dramatically reducing the number of states from one level to the next may not suffice to ensure this condition. One *can* introduce a state transition penalty to enforce a slower rate of state change, but it is difficult to implement this scheme in practice: one often must derive the correct penalty empirically, and it is sometimes difficult to establish the correct rate of state change. Finally, the dependencies expressed by the LHMM are, in a sense, backwards. Denoting the state at level i and time slice t as S_t^i, and the observable at level i and time slice t as O_t^i, the dependencies in the LHHM are:

Graphically, this set of dependencies can be represented as shown in in Figure 5.2.

Note that the state S_t^i is *not* dependent on state S_t^{i+1} as one might expect. The only interaction between levels is in the observables; specifically, the observable O_t^i is dependent on both S_t^{i-1} and S_t^i. Put another way, S_t^{i-1} and S_t^i are mutually conditionally independent given O_t^i. It would be preferable to have a model in which low level states are dependent on high level states; i.e., one in which $S_t^{i-1} \leftarrow S_t^i$. To illustrate this principle, consider an

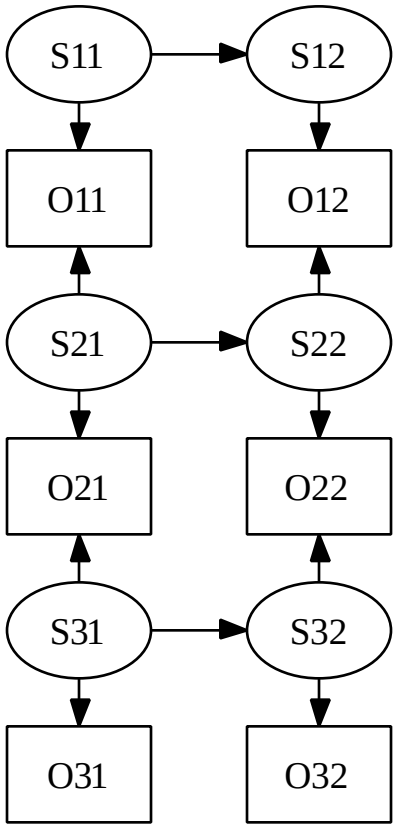

Figure 5.2: Graphical Representation of Layered HMM Dependencies

office setting in which the main audible sound is a human typing on a keyboard. Here, the typing is the *evidence* that there is a human doing work on a computer, and the room is in the typing state *because* there is a human typing on a keyboard. If there happens to be evidence that there is a human in the office, there is no way in the LHMM framework to express the fact that typing is now more likely than if there were no human present in the office absent construction of sub-models which can be traversed.

Another project of note using LHMMs is due to Oliver et. al., who in [77] demonstrated an LHMM system for recognizing human activities in an office setting. Their work, like Clarkson's, used both audio and video information and showed that using both modalities together was superior to using either alone. They also showed the ability to adapt models learned in one office to other offices by retraining only the bottom layer of the LHMM. Finally, they showed that there was essentially no difference between using the full posterior matrix and using only a set of discrete symbols for inter-layer communication.

The Hierarchical HMM

A model which addresses all the problems with the LHMM is the hierarchical HMM (HHMM). In the HHMM, first described by Fine et. al. in [32], each state can emit either an observable *or* a sequence of substates. Structures can be shared in the HHMM; that is, a given substate can appear as a child of multiple high-level states. In addition to states Q and outputs O, the HHMM introduces an extra multilevel variable, F, which controls the rate at which high-level states change by permitting such changes only when child HMMs reach an accepting state. The dependencies between all variables in the HHMM are:
where I is the total number of levels in the HHMM. A graphical representation of these dependencies is given in Figure 5.4. In this model, multiple levels of states can influence the observables, though in practice, dependencies from high-level states to observables are frequently omitted. Unlike the LHMM, each state is directly dependent on the state

$$O_t \leftarrow Q_t^0 \dots Q_t^i, \tag{5.6}$$

$$Q_t^0 \leftarrow Q_{t-1}^0, F_{t-1}^0, \tag{5.7}$$

$$Q_t^i \leftarrow Q_{t-1}^i, F_{t-1}^{i-1}, F_{t-1}^i, \tag{5.8}$$

$$Q_t^I \leftarrow Q_{t-1}^I, F_{t-1}^I, \tag{5.9}$$

$$F_t^0 \leftarrow Q_t^0, Q_t^1, \tag{5.10}$$

$$F_t^i \leftarrow Q_t^i, Q_t^{i+1}, F_t^{i-1}, \tag{5.11}$$

$$F_t^I \leftarrow Q_t^i, F_t^{i-1}; \tag{5.12}$$

Figure 5.3: Hierarchical HMM Dependencies

immediately above it in the hierarchy, an arrangement more reflective of reality than the independence that obtains in the LHMM. Finally, decoding can be accomplished in one reasonably efficient Viterbi pass, as described below. All these features are exactly those which match an intuitive sense of a multiresolution model of the real world, making the HHMM an attractive choice for implementation.

The naive approach to inference in the HHMM was given by Fine et. al. in [32]. Their inference algorithm was a straightforward adaptation of the inside-outside algorithm for parsing with statistical context-free grammars. This inference algorithm, like inside-outside, is cubic in the observable sequence length T (see [57] and [68]), making it too inefficient to use in practice. In [73] and [72], Murphy presented an exact inference algorithm for HHMMs which used the junction tree algorithm, originally developed for inference in dynamic Bayes networks. This algorithm was linear in time; specifically, for a D-level HHMM with Q states, inference on a T-length observable sequence was given as $O(TDQ^{1.5D})$. Murphy further noted that by using approximate DBN inference methods, inference time could be reduced to $O(TDQ^D)$. Xie et. al., in [113] and [114] suggested an

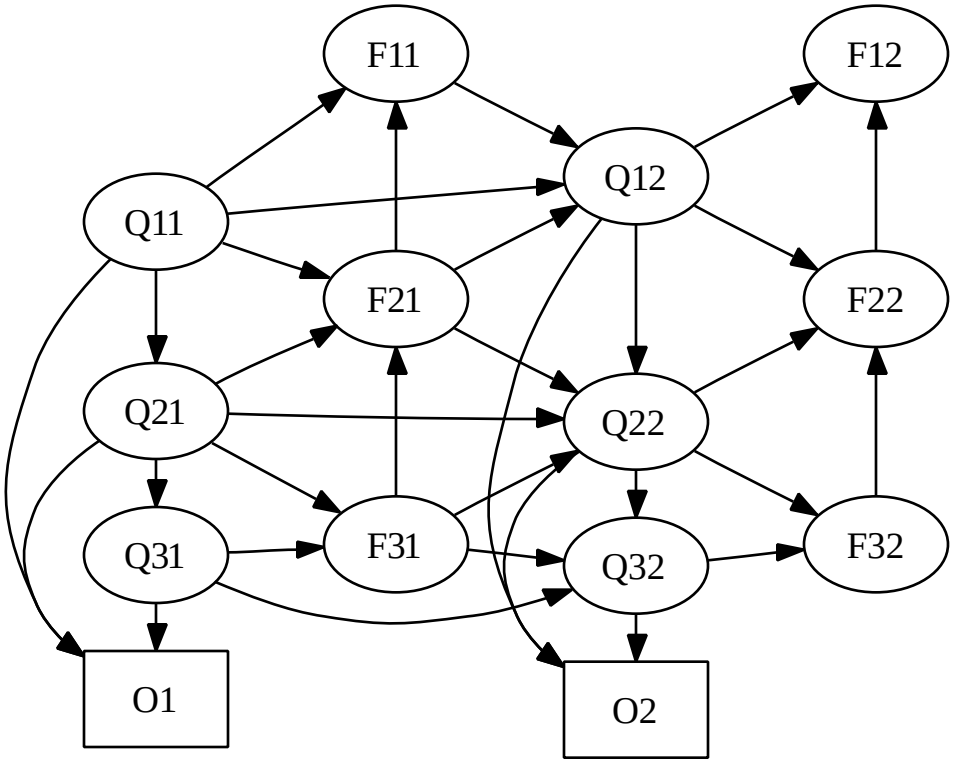

Figure 5.4: Graphical Representation of Hierarchical HMM Dependencies

alternate approach. They advocated compiling the HHMM down into a standard HMM by representing each possible configuration of HHMM states as a single meta-state, allowing for use of the standard Viterbi algorithm with time complexity $O(N^2T)$, where N is the total number of states in the flattened HMM. One issue with this approach is that N may be larger in practice than $DQ^{1.5D}$. Xie et. al. found that inference in practice with the flattening method was $O(TDQ^{2D})$; in other words, the total number of states in the flattened HMM was proportional to DQ^D.

Murphy specifically disrecommended flattening in [73]. The reasons given were:

- A flat HMM cannot provide a multi-scale interpretation of the data.
- Flattening loses modularity, since the parameters of sub-HMMs get combined in a complex way.
- Training HMMs separately and combining requires segmented data.
- A flat HMM may have more parameters. It is the ability to re-use sub-models in different contexts that makes HHMMs more powerful than standard HMMs, since the parameters of such shared sub-models only need to be learned once.

Xie's method seems to address these objections in the following way. First, each flattened HMM state specifically represents one possible HHMM meta-state; hence, multi-scale interpretation is still possible. Second, the dependencies between observables and high-level states are removed, meaning that sub-models can still be trained and reused in the standard way.

An open question is how to learn hierarchical structures from data. Murphy suggested a piecemeal approach in which horizontal structures could be learned by methods commonly

used for standard HMM structure learning; he specifically referred to Brand's work ([9]), in which model structures started out with a large number of states fully connected and parameters were removed according to some optimality criterion. Work by Freitag and McCallum ([36]) and Stolcke and Omohundro ([97]) was also cited in this context. Murphy suggested that vertical relationships could be learned by similar methods, but cautioned that it was not clear how to detect when substructures should be shared. Xie et. al. used a Monte Carlo approach starting with a single state in which, at each iteration, an operation from the set $\{\texttt{split\_state}, \texttt{merge\_states}, \texttt{swap\_children}, \texttt{run\_em}\}$ was selected at random; the resulting model was either kept or discarded according to a BIC ratio. Xie et. al. used this approach to learn 2-level models of soccer videos whose segmentation performance exceeded that of models manually constructed using expert knowledge.

Tangentially related to the problem of structure learning in HHMMs is Pfleger's work ([78], [79]), which sought to learn hierarchical structures online without storage of massive amounts of data. Pfleger's work was aimed at symbolic learning of structure for autonomous, real-world systems with limited storage; his solution involved the estimation of true structure probabilities based on a limited number of observations. His method relied on probabilistically adding structures to the model when observed more than once, and subsequently eliminating rare models, where rarity was judged using Hoeffding bounds. In this work, structures which occur frequently are less likely to lose a Hoeffding race and thus more likely to be kept.

5.4 Structure Learning In HHMMs via Redundancy Reduction

All of the previous approaches given for learning the structure of HHMMs noted above were essentially maximum likelihood approaches; that is, they explicitly seek the model

which maximizes the likelihood of the training data. In some cases, this maximization is subject to constraints dealing with model complexity, like the BIC ratio of Xie et. al. However, maximizing likelihood is not necessarily the optimal metric for organizing a perceptual system. Rather, according to perceptual theory, one should seek an organization in which the redundancy of the system is minimal. In terms of scene learning, the preference for minimal redundancy amounts to finding those sequences or groups of symbols such that by modeling these sequences or groups as coherent units, the redundancy of the data is decreased. This approach is similar to one employed by Ron et. al. in [85] and by Ries et. al. in [84] for structure learning in natural language, with the goal of improving statistical language models.

Given the Markov assumption, there are two ways to minimize redundancy. First, one can seek to minimize the zeroth-order redundancy of the data, which amounts to minimizing the Shannon redundancy (or equivalently, maximizing the entropy) of the data. Shannon redundancy, originally shown in Chapter 2, is repeated here as Equation 5.13. The second way in which redundancy can be minimized is at the first order. That is, the redundancy of the next symbol given the current symbol can be reduced. One minimizes this temporal redundancy by seeking those structures which minimize the average mutual information between symbols at consecutive time indices. This metric, given in Equation 5.14, can be divided by the entropy of the symbol sequence as shown in Equation 5.15 in order to yield a normalized figure.

$$\mathcal{R}_P(X) = 1 - \frac{H_P(X)}{\log |X|}. \tag{5.13}$$

$$I_P(X_{t-1}; X_t) = H_P(X) - H_P(X_t|X_{t-1}). \tag{5.14}$$

$$\hat{I}_P(X_{t-1}; X_t) = 1 - \frac{H_P(X_t|X_{t-1})}{H_P(X)}. \tag{5.15}$$

In what follows, the metric $I_P(X_{t-1}; X_t)$ will be abbreviated $I_P(X)$. Given one of these metrics, one can learn a hierarchical structure in a greedy fashion by iteratively proposing symbol merges and accepting the one yielding the largest reduction in redundancy. A number of constraints can be built into this process, including limiting the number of children a symbol can have, limiting the number of levels in the model, allowing mutable sequence order or requiring fixed sequence order, and forcing the algorithm to exhaust all possible level n possibilities before moving on to level $n + 1$. Collectively, these constraints can have a rather large impact on the final structure that is learned; as shown below, proper choice of constraints is quite important to achieve fitness for some particular task. This learning algorithm, Structure Learning through Redundancy Reduction (SLRR), is given in Figure 5.5.

By way of example, the metric $\hat{I}_P(X)$ was used to learn hierarchical letter sequences from the English Gigaword corpus produced by Graff for the Linguistic Data Consortium [38]. For this experiment, 100,000 words were selected at random from the corpus and case-normalized. After removing spaces and punctuation, the scene learning approach was tested at the character level; there were 413,450 total characters in the data set. There were no constraints placed on the learned structures. Table 5.1 shows the first 50 structures learned, along with their counts in the corpus.

There are several things to note about these structures. First, note the convention that repeated characters are merged into a single symbol; hence, "althexpertsay" is actually a learned structure representing the phrase "all the experts say." Second, note that the structures are hierarchical in that some small structures are re-used in multiple larger structures. For instance, "thexp" is used in both "all the experts say" and "and the explosion." Third, note that, even though the first several structures are very common, the remaining structures on the list have relatively small counts. Given that the corpus segment in question is roughly half the length of Melville's Moby Dick [70], this is a somewhat surprising result. However, recall that the goal is not necessarily to learn the most frequent structures or to

Structure Learning through Redundancy Reduction

Given: Corpus C, set of symbols S, constraints ψ, metric M, set of structures H:
Initialize: Compute $M(C)$.
repeat
    $A \leftarrow \{\}$, $C' \leftarrow C$, $M(C^*) \leftarrow M(C)$.
    for each symbol pair (S_i, S_j) :
    if $\psi(S_i, S_j) =$ TRUE :
      then
        Create new symbol S_q.
        $C' \leftarrow sS_i + S_j + S_q$.
        Compute $M(C')$.
        if $M(C') < M(C^*)$:
          then
            $A \leftarrow (S_i, S_j)$, $C^* \leftarrow C'$,
            $\hat{I}(C^*) \leftarrow \hat{I}(C')$.
if $\hat{I}(C^*) < \hat{I}(C)$:
  then
    Add A to H,
    $C \leftarrow C^*$, $\hat{I}(C) \leftarrow \hat{I}(C^*)$.
  else return

Figure 5.5: Structure Learning through Redundancy Reduction

Rank	Structure	Count	Rank	Structure	Count
1	t+h	5,697	26	althexptert+s	1
2	th+e	2,209	27	althexpterts+a	1
3	v+e	1,446	28	althexptertsa+y	1
4	c+h	647	29	ve+xp	1
5	w+h	625	30	i+vexp	1
6	v+i	483	31	e+althexpertsay	1
7	v+a	168	32	sh+exp	2
8	s+h	825	33	nthexp+e	1
9	j+u	126	34	v+j	2
10	x+p	95	35	o+vj	2
11	e+xp	81	36	g+ovj	1
12	the+xp	6	37	x+ju	1
13	n+thexp	2	38	a+xju	1
14	d+thexp	2	39	axju+s	1
15	n+dthexp	1	40	vi+v	1
16	a+andthexp	1	41	viv+o	1
17	andthexp+l	1	42	vivo+r	1
18	andthexpl+o	1	43	r+vivor	1
19	andthexplo+s	1	44	u+rvivor	1
20	andthexplos+i	1	45	s+urvivor	1
21	l+thexp	1	46	survivor+s	1
22	lthexp+e	1	47	thexp+e	1
23	a+lthexpe	1	48	xp+r	1
24	althexpe+r	1	49	xp+l	1
25	althexper+t	1	50	g+exp	1

Table 5.1: Top structures found in English text by reducing temporal mutual information

maximize the likelihood of the data given some model; rather, it is to discover structures in such a way that the dependency between adjacent symbols is reduced. For this reason, rare structures can be selected provided that they have very high predictive power. Compared to a uniform distribution of letters, even the rare structures are fantastically over-represented. For instance, consider the 31^{st} structure, "ealthexpertsay." The expected count of this sequence in a 413,000 character uniformly distributed text is approximately 1×10^{-16}; the actual count of 1 could thus be considered extremely large. Nonetheless, if one were to propose these structures as the core structures of English text, one would have a hard time defending such a position. Use of the Shannon redundancy metric, however, results in a set of structures that fits this description much better. Table 5.2 shows the first 50 structures learned by reducing the Shannon redundancy metric, along with their counts in the corpus.

These structures look more like what one would expect; there are thirteen common short words in addition to many common prefixes and suffixes. In English text, it is clear that the Shannon redundancy approach is a better way to think about structures than the temporal mutual information approach. It is unclear whether or not this result holds for sensory data.

5.5 Example Application: Interruptibility Estimation for the CHIL Connector Service

An example application which could benefit from multiresolution scene identification is interruptibility estimation for mobile telephones. 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.

Rank	Structure	Count	Rank	Structure	Count
1	e+r	3,722	26	o+m	1,079
2	i+n	4,039	27	i+d	891
3	t+h	5,697	28	in+g	1,517
4	e+s	3,261	29	e+t	788
5	a+n	3,346	30	an+d	1,347
6	t+o	2,449	31	a+c	793
7	e+a	1,817	32	t+s	649
8	e+n	2,582	33	e+w	769
9	e+d	2,264	34	a+m	700
10	a+t	1,897	35	a+y	698
11	o+r	2,010	36	er+s	621
12	i+s	1,705	37	t+i	569
13	a+s	1,665	38	e+m	636
14	th+e	2,209	39	p+r	589
15	o+n	2,453	40	c+h	647
16	a+l	1,881	41	o+l	683
17	a+r	1,597	42	u+r	600
18	i+t	1,301	43	o+w	660
19	o+u	1,316	44	i+r	502
20	i+c	1,114	45	a+d	591
21	e+l	1,212	46	o+s	509
22	e+c	1,005	47	u+s	508
23	s+t	888	48	e+f	527
24	o+f	1,336	49	i+g	528
25	i+l	1,304	50	t+r	446

Table 5.2: Top structures found in English text by reducing Shannon redundancy

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. This alleviation is the specific goal of the CHIL Connector service; introductions to this service can be found in [107], [1], and [25].

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 thesis, the focus is 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.

5.5.1 Prior Work on Interruption

In [88], Siewiorek 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. 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 in-

terrupted 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 [43] 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 achieved promising results in this domain. They further demonstrated in [33] 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 [42] the use of audiovisual sensors for estimating 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 [25], 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 ([29], [30]) and in this thesis, 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, this approach does to some extent capture the relevant information. 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.

5.5.2 An Interruptibility Model

The interruptibility model proposed here encompasses the following variables and sets of dependencies: I represents interruptibility. In principle, humans can judge their own interruptibility as either a real-valued or a binary-valued variable, while a computational system will always represent I as a probability. In this study, human-reported interruptibility levels are binary. 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. In some cases, there may also be available observable visual evidence E_V; in this thesis E_V was used by humans for labeling and so was also considered for modeling. C_H is 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 Equations 5.16 - 5.20.

$$C_E \leftrightarrow C_A, \tag{5.16}$$

$$E_A \leftarrow C_E, C_A, \tag{5.17}$$

$$E_V \leftarrow C_E, C_A, \tag{5.18}$$

$$C_H \leftarrow E_A, E_V, \tag{5.19}$$

$$I \leftarrow C_H, C_O. \tag{5.20}$$

In this work, only the hidden user state information C_H is considered. Both continuous audio data and sequences of still images are used in this work. These two modalities are preferable to a more detailed modality like video for two reasons. First, the task is to detect interruptibility in a *mobile* environment. As such, it is important to recognize that both sensory capacity and computational power available are inherently limited. Most users

would reject a system that required any equipment other than a smartphone; it is thus necessary to forgo 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.

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 some state sequence S^* is inferred from data, the task is to estimate the value of I. More specifically, since this interruptibility estimate will ultimately be combined with estimates from non-sensory modalities, the probability $P(I|S^*)$ must be estimated for all values of I. Using Bayes' rule and borrowing from ASR the engineering convention of weighting the prior and the likelihood, the procedure for estimating $P(i|S^*)$ for some value $i \in I$ is shown in Eqns. 5.21 - 5.24.

$$
\begin{aligned}
P(i|S^*) &\propto P(S^*|i)P(i), && (5.21)\\
&= \prod_{t=1}^{T} P(S_t|i)P(i), && (5.22)\\
&= TP(i)\prod_{t=1}^{T} P(S_t|i), && (5.23)\\
&\approx \alpha P(i)\frac{\beta}{T}\prod_{t=1}^{T} P(S_t|i). && (5.24)
\end{aligned}
$$

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.

As noted above, visual information was considered in addition to auditory information.

Rather than attempting to merge auditory and visual features into a single feature vector, separate estimations of I are carried out for each modality, allowing for late fusion. The fusion procedure is shown in Equation 5.25. This procedure rests on the assumption that the auditory and visual information are independent. This assumption does not always hold; as shown in the dependency model above, both are related to environment and activity. The independence assumption is made in the interest of simplifying the model.

$$P(i|S_A^*, S_V^*) = \lambda P(i|S_A^*) + (1-\lambda)P(i|S_V^*). \tag{5.25}$$

5.6 Interruptibility Experiments Using High-Quality Audio

In [65], experiments on interruptibility estimation using high-quality audio are described. These experiments, which used audio signals collected with a portable audio device in conjunction with images collected periodically with a smartphone camera, show two results: first, that it is possible to estimate user interruptibility in mobile settings using audio, and second, that using the scene learning approach presented above improves performance over both an event-level baseline and an LHMM scene learning approach.

5.6.1 Data Collection

The data used in this study were collected by a single user 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

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 5.3: Evaluation Corpus

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 Table. 5.3. In addition to serving as visual input, the images were also used to label the corpus for interruptibility.

5.6.2 Feature Extraction

From the audio signal, 11 MFCCs were extracted signal at a rate of 100 frames per second. Three additional features were extracted 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, they were filtered with a Gaussian smoothing window. Finally, each feature was normalized globally to zero mean and unity variance.

Visual information was characterized for these experiments by local features and the correlations among local features. Three types of local features were extracted from 4

$\times$ 3 = 12 regular granularities of each image. In each local image patch, the mean of grayscale values, the means of R, G, and B values (the redundant information here is to emphasize grayscale values), and the 12-bin color histogram in HSV color space were used. 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 $\times$ 3 matrix V_{rgb} and the color histogram is denoted 12 $\times$ 12 matrix V_h.

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

$$M_g = V_g V_g^T, \tag{5.26}$$

the RGB mean correlation matrix M_{rgb}:

$$M_{rgb} = V_{rgb} V_{rgb}^T, \tag{5.27}$$

and the color histogram correlation matrix M_h:

$$M_h = V_h V_h^T. \tag{5.28}$$

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

$$F_v = [V_g, V_{rgb}, V_h, M_g, M_{rgb}, M_h]. \tag{5.29}$$

5.6.3 Model Initialization

The scene learning approach presented here is bottom-up; hence, the data must first be segmented into some base or event-level sequence of states. Since manual labeling or

Segmental K-Means

1 Given: k, n, t, f:

2 Initialize: choose k samples S to initialize M.

3 **repeat** until convergence

4 Build grammar g from M.

5 Segment f using g.

6 EM training until parameters converge.

Figure 5.6: The Segmental K-Means Algorithm

general audio is a cumbersome, expensive, and error-prone process, an unsupervised approach is required. There are a number of possible methods to obtain an initial clustering and segmentation given a set of audio data. Three candidates are the segmental k-means (SKM), agglomerative clustering, and a segmental variant of KVKM, k-segment k-means (KSKM). SKM, shown in Figure 5.6, requires the size of the state space to be selected *a priori*, agglomerative clustering, shown in Figure 5.7 requires careful selection of a stopping criterion, and KSKM, a variant of KVKM shown in Figure 5.8, is a leader-follower algorithm which requires careful selection of the model spawning threshold. All of these methods require an initial data segmentation, which can be performed either by some form of change detection (see Chapter 3) or by temporally uniform selection.

For simplicity and speed, SKM is the best choice. In this work, SKM was used with 32 models and 3 states per model. Two additional parameters were set in order to control average event duration after segmentation; the number of frames assigned to initialize each model, and the transition penalty between states. A variety of parameter settings were tested for initial segmentation; the resulting average event lengths are shown in Table 5.4.

As this table shows, event length per state is much more sensitive to transition penalty

Agglomerative 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 5.7: The Agglomerative Segment Clustering Algorithm

Transition Penalty	Initial Frames Per State				
	33	66	100	133	166
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 5.4: Average Event Length in Seconds Per Transition Penalty and Frames Per State Allocation

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 5.8: The K-Segment K-Means Algorithm

than to initial frame assignment. An intuitive understanding of the length of acoustic events led to the selection of the model with a transition penalty of 62.5 and initial frame assignment of 166. Using these settings, SKM produced a segmentation for the entire dataset consisting of 27,432 tokens.

5.6.4 Scene Learning

Using the event-level segmentation as the baseline corpus, multilevel HMMs were built using both the multilevel SKM approach [1] and the SLRR approach using the temporal mutual information metric $I_P(X)$. Using both SKM and SLRR, 4 levels of scenes were learned. In the SKM approach, the number of models per level was halved, and various transition penalties were tested. The resulting average scene lengths are shown in Table 5.5. Note that the transition penalties tested are much smaller than those tested in the event segmentation tests above. This is because the input space for scene learning is monodimensional and discrete rather than multidimensional and continuous; hence, the emission probabilities are roughly the same order of magnitude as the transition probabilities and the transition penalties need not be as high. As this table shows, scene length, like event length, is heavily dependent on the transition penalty. The penalty 0.125 exhibits the smoothest increase in scene duration and was chosen for evaluation.

For the SLRR test, sequential merging with the $\hat{I}_P(X)$ metric was used 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 HHMM level *or*

[1] Note that multilevel SKM does not produce *hierarchical* models, as there is no sense of shared substructure.

Model Level	Number of Models	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 5.5: Average Scene Length Per HMM Level and Transition Penalty, SKM Scene Learning

.

(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 in the corpus to be considered for merge.

3. A minimum of two new merges per level is required; if no proposed merge lowers the redundancy metric, then accept the merge than minimizes it.

The number and average duration of scenes learned, as well as $\hat{I}$ measures at each level, is shown in Table 5.6.

There are several results to note in this table. First is that there were only a total of 23 scenes learned. This rather low total would seem to indicate that there was actually not much short-term predictability in the source corpus that is captured by strict sequential relationships. Second is that most of the scenes learned were themselves very short — the average time spent in any state increases only about 15% from the bottom level to the top.

Model Level	Scenes Learned	$\hat{I}$	Avg 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 5.6: Scenes Learned, $\hat{I}$, and Average Scene Length in SLRR-Trained HHMM

Third is that the temporal mutual information is quite low to start and does not decrease much, which supports the contention that not much short-term temporal structure is present in the corpus. Finally, we note that, unlike a LHMM trained with SKM, an HHMM trained with SLRR exhibits an *increase* in the number of states in the model as more levels are added. There were 48 states in the highest level of the model, compared with only 2 in the LHMM.

5.6.5 Results

One measure of how well the model states should be able to predict interruptibility is to compute the mutual information between interruptibility labels and model states. These figures for the SKM-trained LHMM and the SLRR-trained HHMM are shown in Table 5.7. The single best relationship between interruptibility and model state is actually exhibited by the event-level HMM initialized with SKM. However, note that the strength of the relationship between interruptibility and state deteriorates much more rapidly in the SKM-trained LHMM than in the SLRR-trained HHMM.

After learning these two kinds of multilevel structures, the interruptibility priors $P(I)$ and per-state interruptibility likelihoods $P(S|I)$ were trained and tested using a round-robin

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

Table 5.7: Mutual Information Between States and Interruptibility Labels

procedure. Each day's audio was successively set aside for testing while the other two days were used for training. A prior-only model was tested along with an event-level model and each HMM level. For tests in which both priors and likelihoods were considered, they were considered with equal weight. Results are shown in Table 5.8. As images, hence labels, were available at unequal intervals, results are indexed by both number of segments and by time. Total miss rate is the sum of the false interrupt rate and the false rejection rate. Note also that an actual user experience would differ from these results, as they reflect the implicit assumption that exactly one call is arriving during each test segment.

Using priors only yields an average miss rate of 11.4% by time, all false interrupts. The situation improves by considering likelihoods at the event level; the per-second miss rate is decreased to 8%, balanced between false interrupts and false rejects. Using the SKM-trained LHMM, performance degrades steadily, though a low false interrupt rate at level 3 represents an anomaly in this trend. Best results are obtained by using the level 4 SLRR-trained HHMM, which has an overall miss rate of 6.5%. This miss rate represents a 43% relative improvement over the prior-only baseline and an 18% relative improvement over the event-level HMM. This result indicates that SLRR-trained HHMMs can learn structures that are useful for a specific context-awareness task, and that these structures are superior for this task to an SKM-trained LHMM. More importantly from an application point of view, the notion that interruptibility is inferable from acoustic evidence is validated.

Model Type	Model Level	Miss Rates (Seg)		Miss Rates (Time)	
		False Int	False Rej	False Int	False Rej
Prior	-	30.8%	0.0%	11.4%	0.0%
Events	0	6.4%	13.8%	4.1%	3.9%
SKM-LHMM	1	9.4%	32.9%	5.9%	26.9%
SKM-LHMM	2	37.4%	23.3%	19.4%	17.6%
SKM-LHMM	3	3.7%	32.0%	2.7%	20.7%
SKM-LHMM	4	37.4%	28.2%	19.3%	19.1%
SLRR-HHMM	1	6.4%	12.7%	4.1%	3.9%
SLRR-HHMM	2	6.4%	12.7%	4.1%	3.9%
SLRR-HHMM	3	4.2%	12.7%	3.1%	3.9%
SLRR-HHMM	4	4.2%	11.7%	2.9%	3.6%

Table 5.8: Average Miss Rates, All Conditions

Image-Based Results

After conducting experiments using audio information only, visual information was considered. This use of visual information for this task is natural; many smartphones have onboard cameras with relevant APIs exposed. Further, the human subjects labeled the corpus for interruptibility using the images as activity reminders and in many cases, visual information can be enough to determine user environment and activity. Finally, in many applications, the failure modes of audio and video can be complementary. Image-based interruptibility models were thus trained and evaluated using the same round-robin procedure used for audio.

The visual context model $P(I|S_V)$ was represented by a set of diagonal-covariance Gaussian densities. Since the feature space was 768-dimensional, and there were only 94 images in the database, the most detailed possible model was a single Gaussian per class. Even his model was severely undertrained, meaning that the results which follow are almost certainly not a useful measure of the utility of visual information for the mobile interruptibility estimation task. Results are shown in Table 5.9. The overall miss rate of 52.7% by time is many times worse than the best audio-based miss rates, and even many times worse than the prior. As noted above, though, the lack of data was certainly a serious problem; hence, it is difficult to draw any conclusions from this result.

5.7 Interruptibility Experiments Using Low-Quality Audio

While the high-quality audio results shown above are exciting, they are not necessarily reflective of the capabilities of a portable cellphone. Data was collected for the prior study using several extra pieces of equipment: a personal audio recording device with, on average, poor battery life, an external microphone which must be worn in a prominent loca-

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 5.9: Miss Rates by Time: Image-based Gaussians

tion on the user's clothing, and a portable battery-powered preamp. Setting aside for the moment the issue of cost — which is not trivial — the social stigma of appearing to be recording audio and snapping photos in public is difficult to bear for any reasonable length of time. An alternative to this very visible, expensive, and uncomfortable mode of data collection is to use the portable cellphone itself. This section describes experiments using a real cellphone with low recording quality for fully-automated audiovisual data collection with two human subjects. Other experimental enhancements included the evaluation of the Shannon redundancy metric $\mathcal{R}_P(X)$ in addition to the temporal mutual information metric, evaluation of strictly sequential versus non-sequential scene learning, and a number of different SLRR constraints on the symbol merging procedure. In addition to the straightforward miss rate evaluations as given for the high-quality audio condition, an analysis of incremental learning and cross-user adaptation is given.

5.7.1 Data Collection

As noted above, data collection in this experiment was carried out using a real cellphone. The Nokia 6600 Smartphone [75] was used for this purpose. This model uses the Symbian S60 operating system [98], for which a Python library exposing most of the telephone's

Subject	Total Seconds	Total Images	Total Hours	Recording Hours	Recording Days
A	64970	3460	18.04	26	6
B	31898	1695	8.86	13	2
Total	96868	5155	26.89	39	8

Table 5.10: Data Collected Per Subject

functions is available. This API was used to write a data collection script which captured a 160 $\times$ 120 jpeg-encoded image every 20 seconds while recording 8 kHz, 16-bit audio in 20-second chunks, a total data rate of less than 1 MB per minute. At this rate, a 1 GB MMC or SD card costing less than $50 can hold over 16 hours of data.

Two subjects were recruited to collect data using this telephone. Worn around the user's neck on a lanyard, the device was inconspicuous and allowed for natural human interaction. Subjects were instructed to warn others of the recording if conversation occurred and offer to turn the device off. A summary of data collected in this fashion is shown in Table 5.10. Note that the total time columns refer to actual temporal extent of the recordings, while the recording columns refer to clock hours or calendar days in which recording took place. The round-robin evaluations were organized around recording hours.

5.7.2 Feature Extraction

For this study, two different audio feature sets were tested. The first feature set consisted of 11 MFCCs extracted at a rate of 100 frames per second from the audio signal and normalized to zero mean and unity variance on a per-subject basis. The second set consisted of a 14-dimensional feature vector containing 11 MFCCs plus centroid, diffusion, and snr, exactly as for the high-quality audio experiment. Visual features were extracted as noted above, except that the source images were only 160 $\times$ 120 pixels.

Subject	Feature Space			
	11d MFCCs		14d MFCCs+	
	Tokens	Event Duration	Tokens	Event Duration
A	32,634	1.99s	64,632	1.00s
B	20,035	1.59s	38,556	0.82s

Table 5.11: Initial Segmentation Results Per Subject

5.7.3 Model Initialization

The same model initialization approach was used for this study as for the high-quality audio study. A 32-model, 3-states-per-model HMM was trained for each subject using SKM with an insertion penalty of 62.5 and an initial frame assignment of 166 frames per state. After 10 iterations of SKM, the resulting initial segmentations for both subjects and both feature sets are shown in Table 5.11.

As with the high-quality audio experiment, a baseline performance measure was established using a round-robin procedure for training and testing state conditional interruptibility models. These results, together with the mutual information between model states and interruptibility states, are shown in Table 5.12.

These results indicate that on average, the enhanced MFCC+ feature set leads to better performance than the standard MFCC feature set; the average miss rate for the MFCC set is 24.06% versus 19.70% for the MFCC+ set. For both subjects, much of the improvement came from a decrease in false interrupts: a 60% relative decrease for subject A and a 20% relative decrease for subject B. Interestingly, for each subject, only 24 states using the MFCC+ feature set were actually present in the final segmentation. Given these results, scene learning experiments were carried out using the MFCC+ segmentation as the event-level corpus.

Subject	Feature	I(I;S)	Miss Rate	False Reject Rate	False Interrupt Rate
A	Prior	—	31.67%	0.00%	31.67%
A	MFCC	0.293	21.06%	13.51%	7.55%
A	MFCC+	0.324	16.07%	13.03%	3.03%
B	Prior	—	43.72%	0.00%	43.72%
B	MFCC	0.242	30.18%	9.21%	20.97%
B	MFCC+	0.299	27.12%	10.23%	16.89%

Table 5.12: Baseline low-quality audio interruptibility results

5.7.4 Scene Learning

A number of scene learning experiments were carried out using both the Shannon redundancy metric and the temporal mutual information metric with various sets of constraints. As in the high-quality audio study, the resulting HHMMs were compared to an SKM-trained LHMM. In this study, reduction of states in the LHMM was much less pronounced; instead of halving the number of states with each level increase, it was reduced by 6. In both the LHMM and HHMM, only 3 levels of scenes were learned. Average scene lengths and tokens per level in LHMMs for both subjects are shown in Table 5.13.

A number of HHMMs were also trained using the SLRR method. Both the temporal redundancy and Shannon redundancy metrics were used. Each parent state was allowed to have a maximum of three children, and both ordered and unordered sequences were learned. In ordered sequence learning, structures of the form $a^+b^+ \rightarrow a + b$ are learned, while in unordered sequence learning, structures of the form $(a(a|b)^+)|(b(a|b)^+) \rightarrow a + b$ are learned. In these learning experiments, a minimum of ten scenes were learned for each of three HHMM levels. Thus, for each subject, four three-level HHMMs were learned for a total of 12 different state segmentations with which to train and test interruptibility models.

Subject	Model	States	Tokens	Avg Scene Length
A	HMM	24	64,632	1.00s
A	LHMM-1	18	33,246	1.95s
A	LHMM-2	12	48,323	1.34s
A	LHMM-3	6	18,484	3.51s
B	HMM	24	38,556	0.82s
B	LHMM-1	18	19,090	1.67s
B	LHMM-2	12	10,817	2.94s
B	LHMM-3	6	6,711	4.75s

Table 5.13: Average Scene Length per Model Level, SKM-Trained LHMM

For each subject, the number of tokens and states, along with average scene length, are shown for each segmentation in Table 5.14.

5.7.5 Results

Interruptibility results for all models were obtained using the same round robin procedure detailed above. Results for the event-level HMM and LHMMs are shown for both subjects in Table 5.15. As is the case in earlier experiments, both the statistical relationship between model state and interruptibility and the miss rate degrade gradually with increasing model level. For subject A, the level 3 LHMM's performance degraded to such a degree that the bare prior achieved a better miss rate than the full state-conditional interruptibility model.

HHMM results are shown for subject A in Table 5.16 and for subject B in Table 5.17; a summary of the best-performing models using all approaches is shown in Table 5.18. The results shown here are quite interesting. First, note that in both cases, the best-performing learned model outperforms the prior by a large margin — 15% absolute, 49% relative for

Subject	Model	Metric	Ordered	States	Tokens	Avg Scene Length
A	HMM	——	——-	24	64,632	1.00s
A	HHMM-1	I	yes	32	56,970	1.14s
A	HHMM-2	I	yes	36	52,955	1.22
A	HHMM-3	I	yes	35	49,085	1.32
A	HHMM-1	I	no	27	58,788	1.10s
A	HHMM-2	I	no	23	51,932	1.25s
A	HHMM-3	I	no	21	38,966	1.67s
A	HHMM-1	R	yes	28	57,829	1.12s
A	HHMM-2	R	yes	29	54,643	1.18s
A	HHMM-3	R	yes	29	52,075	1.24s
A	HHMM-1	R	no	26	59,058	1.10s
A	HHMM-2	R	no	28	55,378	1.17s
A	HHMM-3	R	no	31	51,287	1.26s
B	HMM	——	——-	24	38,556	0.82s
B	HHMM-1	I	yes	25	34,759	0.91s
B	HHMM-2	I	yes	23	29,666	1.07s
B	HHMM-3	I	yes	21	27,220	1.17s
B	HHMM-1	I	no	29	32,414	0.98s
B	HHMM-2	I	no	34	31,952	0.99s
B	HHMM-3	I	no	36	31,930	0.99s
B	HHMM-1	R	yes	26	34,080	0.93s
B	HHMM-2	R	yes	30	30,975	1.03s
B	HHMM-3	R	yes	30	30,069	1.06s
B	HHMM-1	R	no	24	36,310	0.87s
B	HHMM-2	R	no	23	34,450	0.92s
B	HHMM-3	R	no	21	33,419	0.95s

Table 5.14: Average Scene Length per Model Level, SLRR-Trained LHMM

Subject	Model	I(I;S)	Total Miss	False Reject	False Interrupt
A	PRIOR	—	31.67%	0.00%	31.67%
A	HMM-0	0.324	16.07%	13.03%	3.03%
A	LHMM-1	0.258	19.68%	14.12%	5.56%
A	LHMM-2	0.188	21.10%	11.34%	9.75%
A	LHMM-3	0.107	31.67%	0.00%	31.67%
B	PRIOR	—	43.72%	0.00%	43.72%
B	HMM-0	0.299	27.12%	10.23%	16.89%
B	LHMM-1	0.250	29.32%	1.22%	28.09%
B	LHMM-2	0.146	29.56%	10.52%	19.04%
B	LHMM-3	0.116	31.11%	8.27%	22.84%

Table 5.15: Miss Rates, SKM-Trained LHMM

subject A, and 18% absolute, 42% relative for subject B. Second, for both subjects, an HHMM system was either the best performer or indistinguihable from the best performance, suggesting that the SLRR scene learning approach is indeed useful for this task. Third, note that for both subjects, the result of learning interruptibility using only the event level HMM is within 2% of the best system. This result suggests that, even though multi-level HMMs yielded the optimal results for this task, it might be advisable to use simpler models in practice if computational resources are scarce. The difference in error type is also of interest. One might expect that false interrupts would occur in proportion to subject uninterruptibility, and that in fact is exactly what was observed in this experiment. Subject A, uninterruptible less than $\frac{1}{3}$ of the time, suffered only 3% false interrupts. Conversely, subject B, uninterruptible more than $\frac{2}{5}$ of the time, suffered nearly 17% false interrupts. Finally, note that the best results on this corpus were much worse on average than for the high-quality experiment. One might immediately attribute this performance drop to a loss

Model	I(I;S)	Total Miss	False Rej	False Int
Shannon Redundancy, Ordered				
HHMM-1	0.384	17.40%	11.88%	5.51%
HHMM-2	0.371	19.67%	14.67%	5.00%
HHMM-3	0.368	19.17%	13.05%	6.12%
Shannon Redundancy, Unordered				
HHMM-1	0.335	17.52%	9.53%	7.98%
HHMM-2	0.313	16.10%	8.26%	7.84%
HHMM-3	0.327	24.37%	4.65%	19.71%
Temporal Mutual Information, Ordered				
HHMM-1	0.142	25.90%	16.86%	9.04%
HHMM-2	0.102	27.13%	6.66%	20.47%
HHMM-3	0.043	30.04%	0.04%	29.60%
Temporal Mutual Information, Unordered				
HHMM-1	0.062	28.87%	7.91%	20.90%
HHMM-2	0.018	31.67%	0.00%	31.67%
HHMM-3	0.006	31.67%	0.00%	31.67%

Table 5.16: Miss Rates, SLRR-Trained HHMM, Subject A

Model	I(I;S)	Total Miss	False Rej	False Int
Shannon Redundancy, Ordered				
HHMM-1	0.310	27.95%	9.26%	18.68%
HHMM-2	0.344	26.13%	9.46%	16.67%
HHMM-3	0.350	25.30%	9.05%	16.25%
Shannon Redundancy, Unordered				
HHMM-1	0.251	28.98%	11.85%	17.13%
HHMM-2	0.194	34.99%	10.97%	24.01%
HHMM-3	0.199	40.16%	14.18%	25.97%
Temporal Mutual Information, Ordered				
HHMM-1	0.185	27.10%	11.69%	15.40%
HHMM-2	0.149	28.74%	10.56%	18.18%
HHMM-3	0.150	28.92%	10.62%	18.29%
Temporal Mutual Information, Unordered				
HHMM-1	0.143	28.67%	13.12%	15.55%
HHMM-2	0.038	43.48%	26.65%	15.82%
HHMM-3	0.037	38.75%	25.47%	13.28%

Table 5.17: Miss Rates, SLRR-Trained HHMM, Subject B

Subject	Model	Type	I(I;S)	Total Miss	False Rej	False Int
A	Prior	—-	—–	31.67%	0.00%	31.67%
A	HMM-0	—-	0.324	16.07%	13.03%	3.03%
A	LHMM-1	—-	0.258	19.68%	14.12%	5.56%
A	HHMM-2	R-uno	0.313	16.10%	8.26%	7.84%
B	Prior	—-	—–	43.72%	0.00%	43.72%
B	HMM-0	—-	0.299	27.12%	10.23%	16.89%
B	LHMM-1	—-	0.250	29.32%	1.22%	28.09%
B	HHMM-3	R-ord	0.350	25.30%	9.05%	16.25%

Table 5.18: Miss Rate Summary

of audio quality; it is intuitively plausible that with lower-quality audio the acoustic model on which the interruptibility model is based would be less able to cluster the audio into truly perceptually similar states. However, note that the prior error rate for the high-quality audio experiment was only 11.4% by time, and the final error rate was approximately half that figure. The relative reduction from prior error to final error is similar across experiments: 49% relative for subject A in the low quality experiment, 42% relative for subject A in the low quality experiment, and 42% relative for subject in the high quality experiment. This consistency of result suggests that the real driver of final error rate may not be audio quality, but the user's prior interruptibility rate.

Image-Based Results

As with the high-quality audio experiments, image results were considered in this set of experiments. Unlike the high-quality experiments, a larger number of images was available for study. There were, on average, 132 images available per training hour. Subject A thus had, on average, 3300 training images and Subject B nearly 1600. This is still not a

Subject	Full Feature Space			Compressed Feature Space		
	False Int.	False Rej.	Total Miss	False Int.	False Rej.	Total Miss
A	12.29%	15.10%	27.40%	20.39%	15.88%	36.28%
B	13.28%	16.23%	29.51%	29.12%	12.04%	41.14%
Avg.	12.62%	15.47%	28.09%	23.26%	14.62%	37.88%

Table 5.19: Image Results for Low-Quality Audio Corpus, Two-Gaussian System

tremendous amount of data however; assuming a 35% interruptible prior, this means that, for example, Subject A had 1155 sample points for 1536 parameters (a mean and a variance for each feature) in the interruptible state and 2145 sample points for 1536 parameters in the non-interruptible state (the numbers for subject B would be 593 and 1102). The visual feature space, after eliminating symmetric values from the correlation matrices, had 426 dimensions. After removing those features with 0 variance, this space was reduced to 390 dimensions. With this large a feature space, the ratio of samples per parameter is still well short of what would be required for a single well-trained diagonal covariance Gaussian density. For this reason, in addition to repeating the single Gaussian experiment from the high-quality audio experiment, another experiment was carried out in which, after normalizing each visual feature to zero mean and unity variance across the entire training set, PCA was used to reduce the feature set to a more manageable figure: 25 dimensions. This much smaller feature space ensures that the Gaussian density parameters are much more well-trained at the cost of a loss of expressiveness in the feature space. However, this loss of expressiveness is on average small: for both subjects, over 95% of the eigenvalue mass was retained. Table 5.19 shows, for each subject, the average miss rates in both the uncompressed and the compressed feature space with using a single Gaussian density per class.

The miss rates shown here are worse than those for low-quality audio, though the bal-

Subject	Full Feature Space			Compressed Feature Space		
	False Int.	False Rej.	Total Miss	False Int.	False Rej.	Total Miss
A	26.58%	4.95%	31.54%	17.95%	6.52%	24.47%
B	31.21%	5.87%	37.09%	34.52%	2.25%	36.78%
Avg.	28.10%	5.25%	33.36%	23.40%	5.11%	28.51%

Table 5.20: Image Results for Low-Quality Audio Corpus, Full Interruptibility Models

ance between false interrupts and false rejects is both closer and more stable than for audio. Further, for both subjects this performance is an improvement over the prior and in subject B's case, performance is actually competitive with the event-level audio HMM. In order to determine if these results could be improved upon, an additional set of experiments was undertaken in which the visual feature space was clustered using the SKM algorithm and used to construct a full conditional interruptibility model $P(S|I)$ just as in the audio experiments. This experiment was carried out for both the full 390-dimensional feature space and the PCA-compressed feature space with six single-Gaussian states. As with the audio experiments, state clustering was carried out using the entire dataset and a separate conditional interruptibility model was constructed for each step in the round-robin evaluation. Results of these experiments, with prior and conditional weights set optimally, are shown in Table 5.20.

These results indicate a 25% relative improvement in miss rate for the PCA system, but a degradation for the full feature space system. The latter result is likely due to severe undertraining; there were not enough samples available in the 390-dimensional feature space to adequately estimate the parameters of one Gaussian, let alone six. The PCA-compressed 6-state HMM has performance comparable to the two-Gaussian full feature space system with a miss rate of approximately 28%. This figure does not approach audio-based performance; it is likely that in order to improve further, an image-based system would re-

quire much more data to train color-based models. Alternatively, it might be appropriate to abandon color-based models entirely and attempt some form of basic object recognition. Neither alternative is very attractive; collecting the vast amounts of data required to train these models well would be cumbersome and expensive, while object recognition is very computationally demanding. One might thus infer that audio is a better choice for this application, even in low-quality conditions.

5.7.6 Incremental Learning

In the experiments shown above, there was a relatively large amount of audio data from which to learn $P(I)$ and $P(S|I)$ for each test. In deployment, it is unlikely that these amounts of data would be available *a priori*; hence it is useful to explore how to learn incrementally during use and how fast performance approaches acceptable levels. In order to answer these questions, an incremental learning procedure was devised using only the event-level HMM segmentations. In this procedure, the data were partitioned into N recording hours, and for each hour i, the miss rate for hours $i+1$ through N was computed with an unadapted model — in this case, a model in which the priors and likelihoods were all uniform and the tiebreaking procedure is to hypothesize uninterruptibility. After computing these unadapted miss rates as a baseline, incremental learning was carried out. For each hour i, segments from hours 0 through $i-1$ were selected at random for adaptation according to a call schedule and learning policy, the models were adapted, and then hours i through N were tested using the updated model. The segments chosen depended on a call schedule which states the probability of receiving a call during any given 20 second period and on the learning policy which states whether or not the model adapts only when the system makes an error or if it can adapt regardless of the outcome.

This incremental learning procedure was carried out 100 times for each subject and results averaged. Three call schedules were used, with per-segment call probabilities of

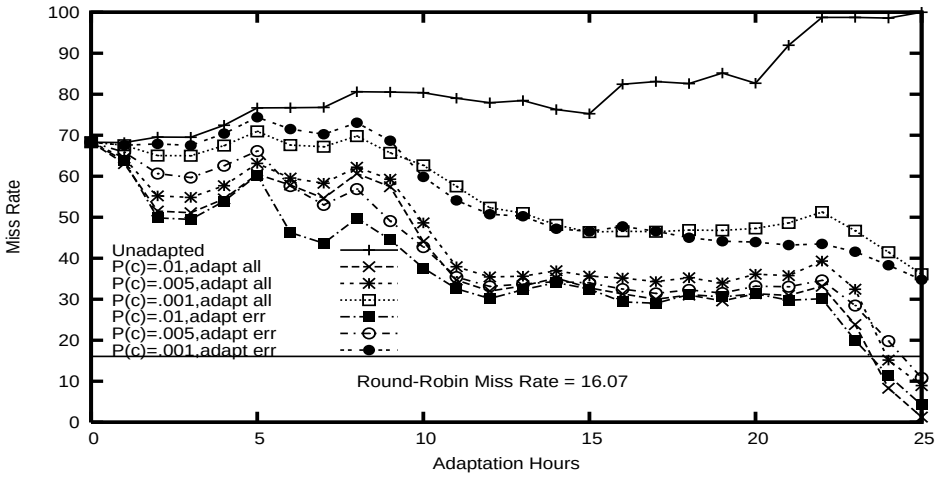

Figure 5.9: Subject A Incremental Learning Performance

.01, 0.005, and 0.001. These figures amount to 40 calls, 20 calls, and 4 calls per 24-hour period. Combined with the two learning policies, there were six incremental learning conditions evaluated per subject. The resulting adaptation curves are shown in Figure 5.9 and Figure 5.11; the average amounts of adaptation time per subject and condition are shown in Figure 5.10 and Figure 5.12. Note that subject A's adaptation curve starts to level out after approximately 15 hours of available adaptation, or 300 seconds of actual adaptation data. Subject B's adaptation performance does not approach round-robin result; however, there were only 13 total hours in this dataset. This result would seem to indicate that at least 15 hours are needed to approach round-robin performance.

5.7.7 Effects of Cross-User Mismatch

A natural question to ask, given the results above, is how well the given approach estimates interruptibility when there is a mismatch between user and model; i.e., how well will the a model trained using subject A's data perform when tested on subject B's data and vice-versa? There are actually three kinds of mismatch to consider. First is a total mismatch, in

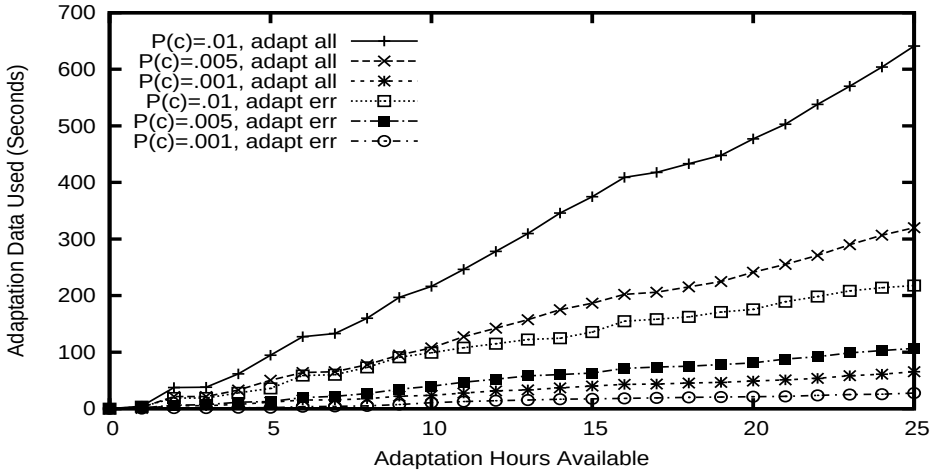

Figure 5.10: Subject A Incremental Learning Time

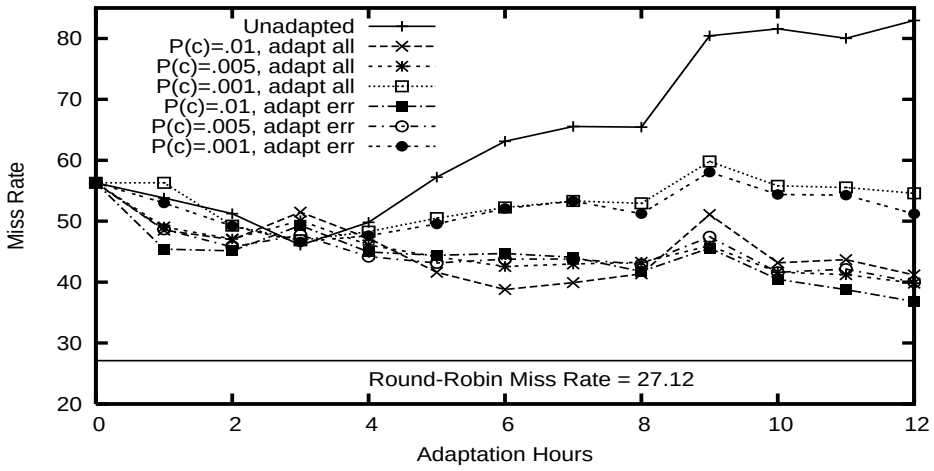

Figure 5.11: Subject B Incremental Learning Performance

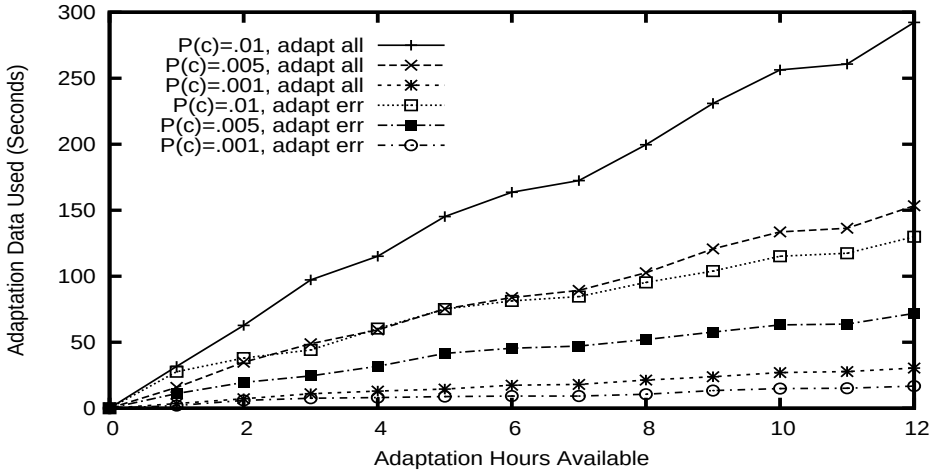

Figure 5.12: Subject B Incremental Learning Time

which both the acoustic model and the interruptibility model are trained on one user and tested on another. Second is an acoustic mismatch, in which the acoustic model is trained on one user and used for segmentation on another user. This new segmentation is now used with the test subject's labels in order to train a matched interruptibility model. Third is an interruptibility mismatch. In this case, a subject's data is segmented with the correct acoustic model, but another user's interruptibility model is used.

In order to evaluate the degree to which performance degrades under these mismatch conditions, cross-user adaptation experiments were carried out using subject A and subject B. The first step was to segment each subject's data using the other subject's acoustic model. The resulting segmentations are compared to the correct segmentations in Table 5.21. Note that for both users, there are many fewer tokens and lower state entropy for the cross-subject case than for the same-user case. This result makes intuitive sense; while there may be some states which are broadly similar across users, it is likely that there may be some fine details which are well-represented in one subject's data but not in the other's. In this case, a same-user model will tend to change states to capture these fine details, while a

Train Subject	Test Subject	Tokens	Avg. Event Length	Entropy
A	A	64,632	1.00	2.98
B	A	63,403	1.02	2.78
B	B	38,556	0.82	3.07
A	B	33,545	0.95	2.64

Table 5.21: Same-user and Cross User Segmentation Results

cross-user model might not.

Each possible mismatch mode was then tested using the HMM-level segmentations with the same round-robin method used above. The results of this experiment are shown in Table 5.22, together with the matched-condition results for comparison. These results are very interesting. For both users, the worst result is to use the correct acoustics but a mismatched interruptibility model. For subject A, this approach did not even beat the prior, while for subject B it beat the prior by only 5% absolute. This result makes perfect sense; a cross-user interruptibility model not only reflects different user preferences, but is trained using a totally different segmentation of the acoustic space. More interesting than this result is what happens when a cross-user acoustic model is used. When the cross-user interruptibility model is used *together* with a cross-user acoustic model, performance degrades, but not very much — 6.5% absolute for subject A, 2% for subject B. When a cross-user segmentation is used to train a new interruptibility model, the results are more impressive still. In this case, subject A's results degrade by less than 2% absolute while subject B's degrade by approximately 0.5%. These results are strongly suggestive of the idea that if an acoustic model is sufficiently broad, it can be employed for more than one user, provided there is some mechanism for producing or adapting interruptibility models online.

Acoustic	Interrupt	Test	Miss Rate	False Rej	False Int
A	A	A	16.07%	13.03%	3.03%
A	B	A	31.67%	0.00%	31.67%
B	B	A	22.63%	8.61%	14.01%
B	A	A	18.01%	9.94%	8.06%
B	B	B	27.12%	10.23%	16.89%
B	A	B	38.84%	0.24%	38.64%
A	A	B	29.35%	2.98%	26.37%
A	B	B	27.72%	11.13%	16.59%

Table 5.22: Subject Mismatch Results, Round-Robin Evaluation

5.8 Chapter Summary

This chapter presented an information-theoretic method for learning multilevel structures from audio data in an unsupervised fashion. This method is based on the perceptual principle of redundancy reduction, and was demonstrated to be effective for a real context awareness task. From an application point of view, a model of interruptibility based on environmental, activity, social factors, and observed human preference was presented. Experiments considering only environmental and activity factors demonstrated that, for high-quality audio, a pleasing level of performance was achievable and that use of the proposed scene learning technique led to a significant improvement in performance compared to both a single-level HMM and a layered HMM. Results on low-quality audio showed a large overall performance degradation compared to high-quality audio. The proposed scene learning technique showed an improvement for one subject in this study of approximately the same absolute magnitude as for the high-quality study. For the other subject, the difference between the single-level HMM and the learned HHMM was essentially zero. For both

subjects, however, the proposed learning technique resulted in better performance than using a layered HMM. Further, the degree to which interruptibility models can be learned incrementally was quantified for a number of combinations of call schedules and learning policies. Given a relatively modest number of calls, the results suggest that at least 15 hours of active use with a cooperative user are required to obtain reasonable performance. Finally, the effects of cross-user model mismatch were quantified. The results of this experiment seem to indicate that interruptibility model mismatch is a more serious problem than acoustic model mismatch. This is a desirable result, as in practice it should be much easier to collect and make use of a simple learning signal (i.e., for incremental learning of the interruptibility model) than to collect and cluster audio from a user in the field. In principle, this result means that as long as one can collect a sufficiently large sample of representative audio from a wide sample of experience types, one ought to be able to deploy a real system whose only concern is to learn an interruptibility model online.

Chapter 6

Conclusions

6.1 Chapter Overview

The contributions of this thesis are both theoretical and practical. These contributions are summarized here, along with a discussion of future directions in machine listening.

6.2 Contributions

This thesis has focused on exploiting auditory information in order to achieve context awareness for real-world computational systems. Using the tools of theoretical machine perception together with engineering paradigms developed for ASR, context awareness systems for three areas of perceptual computing were presented: multimedia gisting, environment recognition, and multiresolution scene learning for mobile interruptibility estimation. These three areas are discussed in turn below.

6.2.1 An Auditory Blind Value Assignment System for Multimedia Gisting Applications

Chapter 3 presented a method for using acoustic information to estimate multimedia segment value for gisting applications without prior knowledge of segment contents or any semantic models of events of interest. Using an information-theoretic measure of signal content as a proxy for value, systems were demonstrated for two specific applications that displayed value judgments broadly similar to those made by human subjects.

Specifically, a system built for after-action review generation achieved a human judgment match rate of 27%, compared to 38.6% for the average human subject. This performance handily exceeds 13.5% for temporally uniform sampling, 1.2% for random sampling, and 16.2% for a video-based system. Further, on segments selected by a large proportion of human subjects, the match rate increased, reaching 43.2% on segments selected by at least 60% of the humans in the study, indicating that system performance on the segments which are more important to match is better than overall performance.

A similar system built for stream-of-interest selection in an online aggregate remote sensing application achieved a human judgment match rate of 36.3%, compared to 41% for the average human subject. This performance exceeds the 25% chance performance, as well as the 26.6% performance achieved by a video-based system. Further, for segments on which humans broadly agreed, the match rate increased, reaching 61.1% on segments with a human agreement level of 0.5 or higher. As with the AARG system above, this result indicates that system performance on segments which are more important to match is better than overall performance.

For both after-action review generation and online aggregate remote sensing, it appeared that human subjects strongly preferred segments containing strong local scene changes. This preference suggests that humans find it difficult to keep track of the entire sensory pastiche and cannot hope to produce value judgments which are optimal in an information-

theoretic sense. Given this interesting observation, it remains an open question whether or not human produced gists convey more information to human consumers than the more theoretically correct automatically generated gists.

6.2.2 An Environment Recognition System for User State Modeling

Chapter 4 presented methods for inferring user environment from acoustic evidence. Using a corpus of environmental audio collected in 9 environmental types on 4 continents, a standard HMM/GMM system was compared to a perceptually-motivated system based on binary trees of linear autoencoding neural networks. Using the 2006 NIST CLEAR evaluation campaign dataset, the best HMM/GMM system achieved an average misclassification rate of 15%; 5% for segments from locales seen during training, and 25% for segments from locales not seen during training. By contrast, the best autoencoder tree system achieved an average misclassification rate of 30% overall; 20% on segments from locales seen during training and 41% for locales not seen during training. Even though the autoencoding method used more than three times as many parameters as the HMM/GMM system, overall performance was not close.

In addition to this performance comparison, human subjects were evaluated as a baseline. On a small subset of the testing corpus, humans achieved an overall misclassification rate of nearly 74%. The HMM/GMM system had a misclassification rate on this subset of 29.6%; 22.2% on segments from seen locales and 37% on segments from unseen locales. While the HMM/GMM performance degradation was unexpected (but also seen for the autoencoding method), performance was still far superior to human performance. This result was not surprising; pilot studies had indicated that humans performed poorly compared to automatic systems on a much more limited task. It is likely that this gap is due largely to a lack of experience in listening tasks, and the perceived oddness of trying to determine environment from auditory rather than visual evidence. Nonetheless, the results

of these environment recognition experiments suggest that this type of system might be useful not only for robot and mobile device context awareness, but also for applications in which human analysts attempt to categorize and extract information from data of unknown origins.

An additional round-robin experiment was conducted in which a GMM system achieved a rather poorer misclassification rate of 50%; it appeared that the data partition in the CLEAR evaluation campaign was particularly easy. To quantify the degree to which labeled data is required to bring performance to acceptable levels, an adaptation experiment was conducted using this round-robin system as a baseline. The result of this experiment was that misclassification rates could be roughly halved with only five minutes of adaptation data, and brought to about 20% with eight minutes in a new locale. This result indicates that, while the ability of the trained system to generalize is weaker than the CLEAR campaign results would indicate, in practice, not much data from a given locale is needed to improve performance.

6.2.3 A Theory of Hierarchical Structure Learning

Chapter 5 presented a theory of structure learning from unlabeled audio based on the perceptual principle of redundancy reduction. Two forms of this approach were presented; one based on reducing the Shannon redundancy (i.e. the unigram redundancy) of the set of symbols in a sensory corpus, the other on reducing the temporal mutual information (i.e. the bigram redundancy). At the core of this approach is the contention that when one symbol is predictable given a previous symbol, these two symbols should be merged into a higher level structure; when all such structures in a corpus have been found, the resulting high level state sequence will exhibit very little redundancy. This method is contrasted with the more standard approach of learning layered HMMs using repeated application of the segmental k-means algorithm. An empirical evaluation of this learning method was conducted

in an indirect fashion, as in general it is difficult to assess structure quality independent of some application. Specifically, this theory of structure learning was used to inform an application intended to estimate user interruptibility from audio for a context-aware mobile telephone task. Experiments carried out in this scenario indicated that hierarchical structure learning is superior to the standard multiresolution approach. This result is largely because the hierarchical learning method does not require *all* low-level events to be merged into high-level scenes as the standard approach does. As a result of this relaxed requirement, a hierarchical structure learned in this fashion may actually have *more* symbols at high levels of modeling, meaning that there is more granularity in the state space and a richer set of possible associations between acoustic states and human states. In two of three cases, the hierarchical learning method also produced superior results to the baseline of a standard one-layer HMM; in the third case results were essentially identical except that the HHMM results were better balanced between false interrupts and false rejections. Finally, it appeared that Shannon redundancy was a better metric for guiding HHMM structure learning than temporal mutual information; examples of structure learning in text using these two metrics given in Chapter 5 provide insight as to why this is the case.

6.2.4 A System for Estimating User Interruptibility for Mobile Applications

In addition to the theoretical contribution of a method for hierarchical structure learning, Chapter 5 also presents a model and system for estimating user interruptibility for mobile applications. This system, which estimates interruptibility by considering observed conditional acoustic state probabilities given interruptibility states, $P(S|I)$, achieved an overall miss rate of 6.5% on a single-subject, high-quality audio database, with a false interrupt rate of 2.9% and a false rejection rate of 3.6%, both measured by time. On a two-subject, low-quality audio database, the system achieved a misclassification rate of approximately

19%; one subject had more false interrupts than rejections (16.25% vs. 9.05%), while the other subject had a more balanced performance profile (8.26% false interrupts, 7.84% false rejections).

The effects of incremental learning were investigated, and it was demonstrated that with a cooperative user, preferences can be learned with relatively few calls. Further, the effects of cross-user acoustic mismatch and interruptibility model mismatch were measured. It was demonstrated that the interruptibility model, which is based on user preference, is much more important than the acoustic model. This observation led to the conclusion that as long as the acoustic model is sufficiently broad, it can be applied to multiple users and incremental learning procedures used to estimate an interruptibility model.

6.3 Future Work

While this thesis presented many interesting results, it also raised several interesting questions which could be investigated in a relatively short timespan. Arranged by topic, some of these questions follow.

6.3.1 Questions in Multimedia Gisting

- Is there a more objective measure of gist quality than human match rate? While attempting to match human performance in gist creation is a reasonable first step, in that it avoids subjective judgments of gist quality, we must not forget that the real goal of gisting is to allow humans to perform some task more easily than they otherwise could. One way to measure gist quality taking this goal into account might be to design some question-answering scenario in which humans given entire recordings are scored against humans given gists created with various methods. While it seems clear from the results in this thesis that humans prefer to tell stories using local scene

change, it is *not* at all clear that their preference is actually operationally useful or correct.

- Can this method be expanded to cover more structured types of multimedia artifacts? One reason this method works well is that naturally occurring data streams are event-sparse. For much of the time in real experience, not much changes or happens. As such, when something *does* change or happen, it can almost automatically be assumed interesting. On the other hand, in more structured multimedia artifacts, like sports or films, the data are much less sparse. Worse, the auditory evidence is often divorced from the events depicted in the multimedia streams — particularly in films which are heavily scored with music. Preliminary, informal explorations indicate that blind value assignment could potentially be used to automatically generate movie trailers, but it is clear that the richness of information in this media type compared to the sparseness in natural media warrants additional investigation.

- Can prior knowledge in some cases be brought to bear in gisting applications? In text summarization, one is often able to "center" the summary on some set of key words or phrases. Likewise, one ought to be able to provide some clues to a multimedia gisting system in order to more easily find the information one wants. Difficulties abound for this approach; for instance, how does one describe a sensory event of interest, and how does a computational system interpret that description? Two possibilities are verbal descriptions and exemplars; both methods could improve the utility of a multimedia data exploration system.

6.3.2 Questions in Environment Recognition

- To what degree are environments really similar or different? The work in this thesis seems to suggest that some environments are harder to recognize as environments

than others. Specifically, airport and train platform seems to be a very difficult environment to recognize computationally, while environments like bus, gallery, restaurant and park seem easy to recognize even across continents. Given this apparent dichotomy, can one characterize in a more formal way which environments can be learned in the general case, and which require particular locale-based knowledge?

- What are the correct features for environment recognition? This work used a combination of MFCCs and a handful of spectral summary features. While this feature set proved superior to competing feature sets, the feature extraction experiments were far from exhaustive. Intuitively, one might expect that qualitative descriptions of the sound field could be useful for this task. For example, expert humans often characterize sound fields along axes like dry vs. reverberant, noisy vs. ordered, loud vs. quiet, open vs. enclosed, etc. Capturing some of these features might prove complementary to the features used in this thesis and might also shed some light about how humans guide their sensory expectations.

- How can personal or universal priors and transitional models be used to improve environment recognition? The system presented in this thesis makes use of five-second segments of audio, presented in isolation. In reality, there will always be more information at hand than this. Previous work by Ellis et. al. and Clarkson seemed to indicate that, even when unsupervised clustering was used, knowledge of priors and likely transitions could help to identify specific locales frequently visited by a user. One might imagine the ability to estimate in fairly short order a transition model between environments which could be built by expert knowledge but whose parameters could be learned online for each user.

6.3.3 A Question in Scene Learning

- What is the proper degree of temporal association for scene learning? In text, it is quite clear that pairs of adjacent words or letters should be the first place to look for structure. Longer-range associations, such as distant n-grams, article-based triggers, or topic associations clearly exist, but much of the useful associations from an engineering point of view still involve adjacency. It is not at all clear that this is the case in sensory data; acoustically detectable events do appear in temporal clusters but often there is some stretch of silence separating them. Further, the events present in these temporal clusters are not always — or perhaps even often — fixed in order. For example, one might consider a typical "working in the office" scene. There are a number of events one might expect to encounter in such a scenario: typing at a computer, talking to a person in the office, talking on the telephone, and the like. But to tie recognition of the scene "working in the office" to a particular order or temporal extent of these events is nonsensical. The scene learning algorithm presented in this thesis, which relied on temporal adjacency, could in principle be improved for sensory data by relaxing the adjacency requirement while maintaining the same basic redundancy reduction approach.

6.3.4 A Question in Interruptibility Modeling

- What is the proper policy for information integration? In the model presented in this thesis, the policy is late fusion and independence of information sources. That is, auditory information is considered independently of visual information, of social information, and of other ancillary data sources like time of day or GPS coordinates. It is assumed that some executive process is collecting interruptibility estimates from all these sources and integrating them in some sane way. Clearly, though, these information sources are not at all independent. One can imagine, for instance, a case

in which a user will accept a call from a funding agent while driving, but from no one else. Conversely, that same user might not accept a call from a funding agent while in an important meeting, but would accept a call from the person delivering pizza to the meeting. In these cases, late fusion would typically fail, while more tightly coupled models might make the correct estimates.

6.4 Challenges for Machine Listening

This thesis represents only a few small steps toward full exploitation of the audio stream for context awareness. Many specific areas still need to be addressed in order to improve the utility of machine listening for context-aware applications. In no particular order, some of the more important of these are:

Acoustic Event Detection in Real Environments As noted in Chapter 5, automatic systems for classification of isolated acoustic events in known environments have achieved reasonable performance levels; error rates are typically on the order of 5% or less. However, the problem of *detecting* acoustic events in an unsegmented acoustic stream remains difficult. None of the participants in the 2006 CLEAR acoustic event detection campaign submitted systems which achieved any sort of acceptable performance; all were in the neighborhood of 100% error, or higher. This increase in error rate is largely to to insertion errors rather than substitutions. There were two reasons for this failure. One was a system design problem; the other a potentially more serious problem.

The engineering issue was that no fielded system was in fact built as a set of isolated detectors. That is, models were trained for every sound class, including nominal silence. These models were connected into a full grammar and used to find a Viterbi path through the entire test recordings. In fact, due to phenomena like sound overlap, speech, and the acoustic-semantic mismatch, this approach has a real handicap from the beginning. A more

sensible approach would be to evaluate each model on a sliding window of audio and hypothesize an event only when the model posterior exceeds some threshold. Unfortunately, these thresholds can only be learned using heldout data, which was in very short supply during the CLEAR campaign.

The second issue is that in many cases, insertion errors can be attributed to the fact that there are many sounds in real environments which are acoustically similar to sounds of interest, but perceptually or semantically different. For instance, if `keys_jingling` is an important acoustic class for a specific application, it is very difficult to build a system which can detect this event in real time *without* falsely detecting jingling coins as a relevant event. Speech is a particular problem, as it contains both harmonic and non-harmonic sounds, and thus often produces spectra which, on short timescales, closely resemble real sounds of interest. One possible solution to this problem is a tighter coupling between acoustic event detection, speaker activity detection, and ASR; such a coupling might help to alleviate the problems presented by speech. At the same time, this approach may not help to alleviate the problems associated with acoustically similar but semantically distinct non-speech sounds, and such coupling would actually require some sort of factorial modeling, as presumably, acoustic events in the environment are relatively independent of speech activity. For these reasons, this area should remain fertile ground for new research.

Discovery of Acoustic Invariants All state-of-the-art ASR systems are based not on words, but on subunits such as phonemes. Each word can be divided into a sequence of phonemes, which can in turn be decomposed into sub-phonemes, and even further into bundles of speech-specific features representing the positions of the speech organs, voicedness or unvoicedness, tone, and the like. In state-of-the-art machine listening systems, however, modeling below the symbol level is still largely ignored. Ellis' concepts of noise cloud, transient, and weft [28] were an early attempt to provide some form of basic acoustic alphabet for the source separation task, but while these three basic sound types shed some

light on that problem, they are largely unused in machine listening systems because they are computationally difficult to detect and they do not provide enough detailed information to distinguish between, say, a gunshot and a hand clap. If instead a set of real-world sub-sound units were to be produced, then modeling real-world sounds could in theory become much easier. As with ASR, data could be shared across high-level symbols, meaning that models would have much more training data and could in principle be more detailed, possibly leading to better performance. As Handel indicated in [41], much of the information that humans use to identify sounds is actually bound up in *temporal* rather than spectral relationships; in other words, it is the rhythms of sound that matter more than specific frequencies. Some attempts have been made to learn temporal acoustic invariants using unsupervised methods; see for example work by Kraft and his colleagues in [56] and [55]. In this work, ICA was used over long windows of analysis frames and the basis functions learned tended to represent temporal auditory phenomena like onsets, offsets, steady state energy, and transients. Use of these bases improved the classification of sounds in a kitchen environment compared to standard acoustic features; this work could be revisited in order to make further progress toward useful acoustic event detection.

Multi-Sensor Audition All the experiments described in this thesis, and by many other researchers, use only a single audio channel. Biological listeners often exploit stereo listening in order to make distinctions that cannot be made with a single channel. Source localization is one result of a two-sensor strategy; combined with knowledge about locations of sources in a room or other environment, this kind of information can be a powerful cue as to the identity of a sound. Multi-sensor input can also be used to estimate reverberations, which could be a useful feature for environment recognition, and to track moving sources.

Marriage of Factorial and Hierarchical Modeling The acoustic signal reaching any sensor is the result of the interactions of the sources and the environment. In any environment, there are potentially many sources emitting acoustic information at any given time. Most of these sources are independent of each other, though some are only *conditionally* independent given the environment. For example, in an office environment, the sound of a telephone ringing and the sound of a keyboard clicking are independent of each other, though there are strong dependencies between those events and the office setting. A hierarchical acoustic model can capture the conditional independence of events given environment, but fails to capture their temporal independence. Source separation is one possible avenue of advance, but factorial modeling is also a promising area, provided the concept of hierarchical dependencies is maintained. One possibility is a context-switching factorial model in which each context is represented by a state which can emit not just a sequence of subordinate states (i.e. a sub-HMM) as in an HHMM, but potentially many sequences of subordinate states (i.e. a sub-FHMM). Inference, parameter estimation, and structure learning for such a model would be at least as hard for this type of model as for an FHMM, making it a potentially interesting problem for the graphical modeling community.

6.5 Thesis Summary

This thesis began with the twin claims that machine listening technologies can provide context awareness in real-world computational systems, and that the design of these systems could be informed by adaptation of general principles of perception, namely redundancy reduction, to specific problem areas. The systems and experiments presented within constitute strong support for these claims.

The first claim, that machine listening technologies can be used to provide context information for real (or potentially real) applications, is supported by the results shown for environment classification, multimedia gisting, and user interruptibility estimation. The

second claim, that redundancy reduction can be used successfully as a general design principle for real (or potentially real) applications in the machine listening area (as opposed to more CASA-like systems), is supported by the results shown for multimedia gisting and user interruptibility estimation. Taken together, the evidence presented in this thesis constitutes a strong argument for the overarching claim that some measure of context awareness can be achieved in the real-world applications of the near future by exploiting auditory information using systems combining basic concepts from perceptual theory with traditional engineering approaches.

Bibliography

[1] The CHIL project. `http://www.chil.server.de`.

[2] J. Atick. Could information theory provide an ecological theory of sensory processing? *Network: Computation in Neural Systems*, 1992.

[3] F. Atteneave. Information aspects of visual perception. *Psychological review*, 1954.

[4] H. Barlow. Sensory mechanisms, the reduction of redundancy, and intelligence. In *National Physical Laboratory Symposium Number 10, The Mechanization of Thought Processes*, 1959.

[5] H. Barlow. Possible principles underlying the transformation of sensory messages. In W.A. Rosenbluth, editor, *Sensory Communication*. MIT Press, 1961.

[6] A. J. Bell and T. J. Sejnowksi. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6), 1995.

[7] A. J. Bell and T. J. Sejnowski. Learning the higher-order structure of a natural sound. *Network: Computation in Neural Systems*, 1996.

[8] A.J. Bell and T.J. Sejnowksi. The 'indepdendent components' of natural scenes are edge filters. *Vision Research*, 1997.

[9] M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 1999.

[10] A. Bregman. *Auditory Scene Analysis*. MIT Press, 1990.

[11] G. Brown. *Computational auditory scene analysis: A representational approach*. PhD thesis, University of Sheffield, 1992.

[12] D. Chen, R. Malkin, and J. Yang. Multimodal detection of human interaction events in a nursing home environment. In *Proceedings of the International Conference on Multimodal Interfaces*, 2004.

[13] D. Chen, J. Yang, R. Malkin, and D. Watclar. Detecting social interaction of elderly in a nursing home evironment. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2006.

[14] S. Chen and P. Gopalakrishnan. Speaker, environment, and channel change detection and clustering via the bayesian information criterion. In *DARPA Speech Recognition Workshop*, 1998.

[15] S. Chu, S. Narayanan, C. Kuo, and M. Mataric. Where am I? Scene recognition for mobile robots using audio features. In *Proceedings of 2006 IEEE ICME*, 2006.

[16] B. Clarkson. *Life Patterns: Structure from Wearable Sensors*. PhD thesis, MIT, 2002.

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

[18] 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.

[19] B. Clarkson and A. Pentland. Framing through peripheral perception. In *Proceedings of the International Conference on Image Processing*, 2000.

[20] B. Clarkson, N. Sawhney, and A. Pentland. Auditory context awareness via wearable computing. In *Proceedings of the Perceptual User Interfaces Workshop*, 1998.

[21] P. Comon. Independent component analysis: a new concept? *Signal Processing*, 36:287–314, 1989.

[22] M. Cooke. *Modeling auditory processing and organization*. PhD thesis, University of Sheffield, 1991.

[23] M. Cooper, J. Foote, J. Adcock, and S. Casi. Shot boundary detection via similarity analysis. In *Proceedings of the TRECVID Workshop*. 2003.

[24] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

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

[26] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[27] R. Duda, R. Lyon, and M. Slaney. Correlograms and the separation of sounds. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, 1990.

[28] D. Ellis. *Prediction-drived Computational Auditory Scene Analysis*. PhD thesis, MIT, 1996.

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

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

[31] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics and Speech Signal Processing*, 32(6), 1984.

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

[33] J. Fogarty, S. Hudson, and J. Lai. Examining the robustness of sensor-based statistical models of human interruptibility. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2004.

[34] J. Foote. Visualizing music and audio using self-similarity. In *Proceedings of ACM Multimedia*, 1999.

[35] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the International Conference on Multimedia and Expo*, 2000.

[36] D. Freitag and A. McCallum. Information extraction with structures learned by stochastic optimization. In *Proceedings of AAAI*, 2000.

[37] R. Gaborski. Automatic detection of novel scenes in video. In *University Technology Showcase*, 2005.

[38] D. Graff. The English Gigaword corpus, 2003. `http://www.ldc.upenn.edu/Catalog`.

[39] G. Gui and S. Li. Content-based audio classification and retrieval using svm learning. Invited Talk, IEEE Pacific Rim Conference on Multimedia, 2000.

[40] G. Guo and S. Li. Content-based audio classification. *IEEE Transactions on Neural Networks*, 14(1), 2003.

[41] S. Handel. *Listening: An Introduction to the Perception of Auditory Events*. MIT Press, 1989.

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

[43] 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.

[44] J. Hurri and A. Hyvärinen. Temporal coherence, natural image sequences, and the visual cortex. In *Advances in Neural Information Processing Systems*, 2002.

[45] J. Hurri and A. Hyvärinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3):663–691, 2003.

[46] J. Hurri and A. Hyvärinen. Temporal and spatiotemporal coherence in simple-cell responses: a generative model of natural image sequences. *Network: Computation in Neural Systems*, 14(3):527–551, 2003.

[47] A. Hyvärinen, J. Hurri, and J. Väyrynen. Bubbles: a unifying framework for low-level statistical properties of natural image sequences. *Journal of the Optical Society of America A*, 20(7):1237–1252, 2003.

[48] A. Hyvärinen and E. Oja. The FastICA MATLAB package. `http://www.cis.hut.fi/projects/ica/fastica`.

[49] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[50] L. Itti and P. Baldi. A principled approach to detecting surprising results in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[51] L. Itti and P. Baldi. A surprise theory of attention. In *Proceedings, Vision Science Society Annual Meeting*, 2005.

[52] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[53] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *Proceedings of the International Conference on Computer Vision*, 2003.

[54] A. Kapoor and S. Basu. The audio Epitome: a new representation for modeling and classifying auditory phenomena. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[55] F. Kraft. Continuous audio object recognition. Master's thesis, Universität Karlsruhe, 2005.

[56] F. Kraft, R. Malkin, T. Schaaf, and A. Waibel. Temporal ICA for classification of acoustic events in a kitchen environment. In *Proceedings of ICSLP-Interspeech*, 2005.

[57] K. Lari and S. Young. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 1990.

[58] M.S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 2002.

[59] M.S. Lewicki and T.J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 2000.

[60] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 2001.

[61] B. Lucas and T. Kanade. An iterative registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981.

[62] J. Makhoul. Linear prediction: a tutorial review. *Proccedings of the IEEE*, 63(4), 1975.

[63] R. Malkin. Context awareness through machine listening. Phd thesis proposal, Carnegie Mellon University, 2002.

[64] R. Malkin. The CLEAR 2006 CMU acoustic environment classification system. In *Proceedings of the 2006 CLEAR Evaluation Workshop*, 2006.

[65] R. Malkin, D. Chen, J. Yang, and A. Waibel. Multimodal estimation of user interuptibility for smart mobile telephones. In *Proceedings of the International Conference on Multimodal Interfaces*, 2006.

[66] R. Malkin, D. Macho, A. Temko, and C. Nadeu. First evaluation of acoustic event classification systems in the chil project. Presentation, Joint Workshop on Hands-Free Speech Communication and Microhpone Arrays (HSCMA), 2005.

[67] 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.

[68] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[69] T.M. Martinez and K.J. Schulten. *Artificial Neural Networks*, chapter A "Neural-Gas" Network Learns Topologies. North-Holland, 1991.

[70] H. Melville. *Moby-Dick*. Richard, 1851.

[71] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference, and Learning*. PhD thesis, University of California Berkeley, 2002.

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

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

[74] A. Ng., M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in NIPS*.

[75] The Nokia 6600 smartphone. `http://www.nokiausa.com/phones/6600`.

[76] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. In Martinez-Conde, Macknik, Martinez, Alonso, and Tse, editors, *Progress in Brain Research*, volume 155, chapter 2. Elsevier B. V., 2006.

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

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

[79] K. Pfleger. On-line cumulative learning of hierarchical sparse n-grams. In *Proceedings of the International Conference on Digital Libraries*, 2004.

[80] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 7(2), 1989.

[81] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[82] The RealPlayer website. `http://www.real.com`.

[83] 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.

[84] 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.

[85] 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.

[86] C. Siagian and L. Itti. Gist: A mobile robotics application of context-based vision in outdoor environment. In *Proceedings of the IEEE CVPR Workshop on Attention and Performance in Computer Vision*, 2005.

[87] M. Siegler, U. Jain, B. Raj, and R. Stern. Automatic segmentation, classification and clustering of broadcast news. In *Proceedings of the DARPA Speech Recognition Workshop*, 1997.

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

[89] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. In *Proceedings of the International Conference on Multimedia and Expo*, 2002.

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

[91] M. Slaney, D. Ponceleon, and J. Kaufman. Temporal events in all dimensions and scales. In *International Conference on Computer Vision Event Workshop*.

[92] M. Slaney, D. Ponceleon, and J. Kaufman. Multimedia edges: Finding hierarchy in all dimensions. In *Proceedings of the International Conference on Multimedia*, 2001.

[93] P. Smaragdis. *Redundancy reduction for computational audition, a unifying approach*. PhD thesis, MIT, 2001.

[94] The WC3 SMIL specification. `http://www.w3.org/AudioVideo`.

[95] E. Smith and M.S. Lewicki. Efficient coding of acoustic structure using a spike timing code. In *Computational and Systems Neuroscience*, 2004.

[96] E. Smith and M.S. Lewicki. Efficient auditory coding. *Nature*, 439(7079), 2006.

[97] A. Stolcke and S. Omohundro. Hidden Markov model induction by Bayesian model merging. *Neural Information Processing Systems*, 5, 1992.

[98] The Symbian operating system. `http://www.symbian.com/`.

[99] A. Temko, R. Malkin, and C. Zieger. The 2006 CLEAR/CHIL acoustic event classification and detection evaluation. In *Proceedings of the 2006 CLEAR Evaluation Workshop*, 2006.

[100] A. Temko, E. Monte, and C. Nadeu. Comparison of sequence discriminant support vector machines for acoustic event classification. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[101] A. Temko and C. Nadeu. Classification of meeting-room acoustic events with support vector machines and variable-feature-set clustering. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[102] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. Technical report, MIT, 2003.

[103] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14, 2003.

[104] A. Torralba, A. Oliva, M. Castelhano, and J. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113(4), 2006.

[105] A. Vaingankar, V. Chajoi, R. Gaborski, and A. Teredesai. Cognitively motivated habituation for novelty detection in video. In *NIPS Workshop on Open Challenges in Cognitive Vision*, 2003.

[106] A. Waibel and K. Lee, editors. *Readings in speech recognition*. Morgan Kaufmann, 1990.

[107] A. Waibel, H. Steusloff, R. Stiefelhagen, and the CHIL Project Consortium. CHIL: Computers in the human interaction loop. In *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[108] D. Wang and X. Liu. Scene analsysis by integrating primitive segmentation and associative memory. Technical report, Ohio State University, 2000.

[109] D. Wang and D. Terman. Locally excitatory globally inhibitory oscillatory networks. *IEEE Transactions on Neural Networks*, 6(1), 1995.

[110] M. Westphal. *Robust kontinuierliche Spracherkennung für mobile Informationssysteme*. PhD thesis, Universität Karlsruhe, 2000.

[111] M. Westphal and A. Waibel. Model-combination-based acoustic mapping. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[112] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. In *IEEE Multimedia*, 1996.

[113] 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.

[114] 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.

[115] W. A. Yost. *Fundamentals of Hearing*. Academic Press, 2000.

[116] T. Zhang and C. Kuo. Heuristic approach for audio data segmentation and annotation. In *Proceedings of the ACM International Conference on Multimedia*, 1999.