Environmental Sound Recognition With Time–Frequency Audio Features

Article *in* IEEE Transactions on Audio Speech and Language Processing · September 2009

DOI: 10.1109/TASL.2009.2017438 · Source: IEEE Xplore

3 authors:

Selina Chu
NASA
24 PUBLICATIONS **1,211** CITATIONS
SEE PROFILE

Shrikanth S Narayanan
University of Southern California
869 PUBLICATIONS **13,738** CITATIONS
SEE PROFILE

C.-C. Jay Kuo
University of Southern California
1,257 PUBLICATIONS **18,157** CITATIONS
SEE PROFILE

Environmental Sound Recognition With Time–Frequency Audio Features

Selina Chu, *Student Member, IEEE*, Shrikanth Narayanan, *Fellow, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—The paper considers the task of recognizing environmental sounds for the understanding of a scene or context surrounding an audio sensor. A variety of features have been proposed for audio recognition, including the popular Mel-frequency cepstral coefficients (MFCCs) which describe the audio spectral shape. Environmental sounds, such as chirpings of insects and sounds of rain which are typically noise-like with a broad flat spectrum, may include strong temporal domain signatures. However, only few temporal-domain features have been developed to characterize such diverse audio signals previously. Here, we perform an empirical feature analysis for audio environment characterization and propose to use the matching pursuit (MP) algorithm to obtain effective time–frequency features. The MP-based method utilizes a dictionary of atoms for feature selection, resulting in a flexible, intuitive and physically interpretable set of features. The MP-based feature is adopted to supplement the MFCC features to yield higher recognition accuracy for environmental sounds. Extensive experiments are conducted to demonstrate the effectiveness of these joint features for unstructured environmental sound classification, including listening tests to study human recognition capabilities. Our recognition system has shown to produce comparable performance as human listeners.**

***Index Terms*—Audio classification, auditory scene recognition, data representation, feature extraction, feature selection, matching pursuit, Mel-frequency cepstral coefficient (MFCC).**

I. Introduction

RECOGNIZING environmental sounds is a basic audio signal processing problem. Consider, for example, applications in robotic navigation, assistive robotics, and other mobile device-based services, where context aware processing is often desired or required. Human beings utilize both vision and hearing to navigate and respond to their surroundings, a capability still quite limited in machine processing. Many of today's robotic applications are dominantly vision-based. When employed to understand unstructured environments [1], [2] (e.g., determining interior or exterior locations [3], [4]), their robustness or utility will be lost if the visual information is compromised or totally absent. With the loss of sight, a vision-based robot might not be able to recover from its displacement. Knowing the context provides an effective and efficient way to prune out irrelevant scenarios. There have been recent interests in finding ways to provide hearing for mobile robots [5], [6] so as to enhance their context awareness with audio information. Other applications include those in the domain of wearables and context-aware applications [7], [8], e.g., in the design of a mobile device such as a cellphone that can automatically change the notification mode based on the knowledge of user's surroundings, like switching to the silent mode in a theater or classroom [7] or even provide information customized to user's location [9].

By audio scenes, we refer to a location with different acoustic characteristics such as a coffee shop, park, or quiet hallway. Differences in acoustic characteristics could be caused by the physical environment or activities of humans and nature. To enhance a system's context awareness, we need to incorporate and adequately utilize such audio information. A stream of audio data contains a significant wealth of information, enabling the system to capture a semantically richer environment on top of what the visual information can provide. Moreover, to capture a more complete description of a scene, the fusion of audio and visual information can be advantageous, say, for disambiguation of environment and object types. Audio signals could be obtained at any moment when the system is functioning in spite of challenging external conditions such as poor lighting or visual obstruction. Besides, they are relatively cheap to store and compute than visual signals. To use any of these capabilities, we have to determine the current ambient context first. Thus, the determination of the ambient context using audio is the main concern of this research.

Research in general audio environment recognition has received some interest in the last few years [10]–[14], but the activity is considerably less compared to that for speech or music. Automatic unstructured environment characterization is still in its infancy. Some areas of nonspeech sound recognition that have been studied to various degrees are those pertaining to recognition of specific events using audio from carefully produced movies or television tracks [15], [16]. Others include the discrimination between musical instruments [17], [18], musical genres [19], and between variations of speech, nonspeech and music [20]–[22]. To date, only a few systems have been proposed to model raw environmental audio without pre-extracting specific events or sounds. In this paper, our focus is not in the analysis and recognition of discrete sound events, but rather

Manuscript received March 10, 2008; revised February 06, 2009. Current version published June 26, 2009. This work was supported in part by the National Science Foundation, in part by the Department of Homeland Security (DHS), and in part by the U.S. Army. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sylvain Marchand.

Selina Chu is with the Department of Computer Science and Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: selinach@sipi.usc.edu).

Shrikanth Narayanan and C.-C. Jay Kuo are with the Ming Hsieh Department of Electrical Engineering, Department of Computer Science and Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089-2564 USA (fax: 213-740-4651, e-mail: shri@sipi.usc.edu; cckuo@sipi.usc.edu).

Digital Object Identifier 10.1109/TASL.2009.2017438

on characterizing the general acoustic environment types as a whole. For readers interested in recognition of discrete sound effects and specific audio events, we refer them to representative work by [15] and [16].

As with most pattern recognition systems, selecting proper features is key to effective system performance. Audio signals have been traditionally characterized by Mel-frequency cepstral coefficients (MFCCs) or some other time–frequency representations such as the short-time Fourier transform and the wavelet transform. The filterbanks used for MFCC computation approximates some important properties of the human auditory system. MFCCs have been shown to work well for structured sounds such as speech and music, but their performance degrades in the presence of noise. MFCCs are also not effective in analyzing noise-like signals that have a flat spectrum. Environmental audio contain a large and diverse variety of sounds, including those with strong temporal domain signatures, such as chirpings of insects and sounds of rain that are typically noise-like with a broad flat spectrum that may not be effectively modeled by MFCCs. In this work, we propose to use the matching pursuit (MP) algorithm to analyze environmental sounds. MP provides a way to extract time–frequency domain features that can classify sounds where using frequency-domain only features (e.g., MFCCs) fail. The process includes finding the decomposition of a signal from a dictionary of atoms, which would yield the best set of functions to form an approximate representation.

The MP algorithm has been used in a variety of applications, such as video coding [23] and music note detection [24]. MP has also been used in music genre classification [25] and classification of acoustic emissions from a monitoring system [26]. In our proposed technique, MP is used for feature extraction in the context of environmental sound [27]. We investigate a variety of audio features and provide an empirical evaluation on 14 different environment types. It is shown that the most commonly used features do not always work well with environmental sounds while the MP-based features can be used to supplement traditional frequency domain features (MFCC) to yield higher automatic recognition accuracy for environmental sounds.

The rest of this paper is organized as follows. Some relevant previous work is discussed in Section II. and a review of different audio feature extraction methods is given in Section III. The MP algorithm is described and MP-based features are presented in Section IV. Section V contains experimental evaluation and empirical comparison of selected features. Section VI presents results of a listening test for studying human abilities recognizing acoustic environments, similar to those used in the automatic recognition experiments. Finally, concluding remarks and future research directions are given in Section VII.

II. Review of Previous Work

As compared to other areas in audio such as speech or music, research on general unstructured audio-based scene recognition has received little attention. To the best of our knowledge, only a few systems (and frameworks) have been proposed to investigate environmental classification with raw audio. Sound-based situation analysis has been studied in [11], [13] and in [8], [28], for wearables and context-aware applications. Because of randomness, high variance, and other difficulties in working with environmental sounds, the recognition rates fall rapidly with increasing number of classes; representative results show recognition accuracy limited to around 92% for five classes [5], 77% for 11 classes [12], and approximately 60% for 13 or more classes [11], [13].

The analysis of sound environments in Peltonen's thesis [13], which is closest to our work, presented two classification schemes. The first scheme was based on averaging the band-energy ratio as features and classifying them using a K-nearest neighborhood (kNN) classifier. The second uses MFCCs as features and a Gaussian mixture model (GMM) classifier. Peltonen noticed the shortcomings of MFCCs for environmental sounds and proposed using the band-energy ratio as a way to represent sounds occurring in different frequency ranges. Both of these experiments involved classifying 13 different contexts or classes. The classifiers and types of features compared were similar to our experiments, but the actual type of classes were different. Similar to their work, we also compared a variety of different class types. In a subsequent paper by Eronen *et al.* [11], they extended the investigation to audio-based context recognition by proposing a system that classifies 24 individual contexts. They subdivided 24 contexts into six higher-level categories, with each category consisting of four to six contexts. Peltonen *et al.* also performed a listening test and reported the findings in [13]. Subjects were presented with 34 samples, each one minute in duration, for the first experiment and 20 samples, of three minutes each, in the second experiment. The tests were mostly conducted in a specialized listening room. Their listening experiment setup is different than the one presented in this work, most notably in how the data were presented to the subjects. The samples used in our study are the same 4 s segments as used in our automatic classification system (details are given in Section VI).

The work by Aucouturier *et al.* [14] also investigated on environmental type of sounds. Their focus is mainly to study the differences between urban environments, or as the authors refer to as urban soundscapes, and polyphonic music. In their system, they propose to model the distribution of MFCCs using 50-component GMMs and to use Monte Carlo approximation of the Kullback–Leibler distance to determine the similarities between urban and musical sounds. They studied the temporal and statistical homogeneity of each of these classes and demonstrated differences in the temporal and statistical structure for soundscapes and polyphonic music signals. However, instead of defining four general classes of urban sounds, (*viz.*, avenue, calm neighborhood, street markets, and parks.), they consider each location as a single class. For example, a specific street (or location) would be considered a class of its own. In contrast, our approach is to consider different streets (or different locations of similar environment) to be of the same class and propose features that furthers generalization.

There has also been some prior work on using matching pursuit for analyzing audio for classification but quite limited. The proposed approach by Ebenezer *et al.* [26] demonstrated the use of MP for signal classification. Their framework classified acoustic emissions using a modified MP algorithm in an actual

acoustic monitoring system. The classifier was based on a modified version of the MP decomposition algorithm. For each class, appropriate learning signals were selected, the time- and frequency-shifting of these signals forms their dictionary. After the MP algorithm terminates, the net contribution of correlation coefficients from each class is used as the decision statistic, where the one that produces the largest value is the chosen class. They demonstrated an overall classification rate of 83.0% for the 12-class classification case. However, the type and nature of the sound classes are unclear since the test data were proprietary (the classes were only identified by their numbers in the report). Another system using MP was presented by Umapathy *et al.* [25]. In this paper, they proposed a technique that uses an adaptive time–frequency transform algorithm, which is based on MP with Gaussian functions. Their work is most similar to our proposed technique in utilizing the parameters of their signal decomposition to obtain features for classification. However, their parameters to the decompositions were conducted with octave scaling and was used to generate a set of 42 features over three frequency bands. These features were then analyzed for the classification of six-class music genres using the linear discriminant analysis (LDA) and were able to achieve an overall correct classification rate of 97.6%.

Our goal in this paper is to study different unstructured environmental sounds in a more general sense and to use MP to learn the inherent structures of each type of sounds as a way to discriminate the various sound classes.

III. Background Review

Several major feature extraction techniques for audio signal processing are reviewed in Section III-A. Then, signal representation using the MP process is discussed in Section III-B.

A. Audio Features

One major issue in building an automatic audio recognition system is the choice of proper signal features that are likely to result in effective discrimination between different auditory environments. Environmental sounds in general are unstructured data comprising of contributions from a variety of sources, and unlike music or speech, no assumptions can be made about predictable repetitions nor harmonic structure in the signal. Because of the nature of unstructured data, it is difficult to form a generalization to quantify them. Due to the inherent diverse nature, there are many features that can be used, or are needed, to describe audio signals. The appropriate choice of these features is crucial in building a robust recognition system. Here, we examine some of the commonly used audio signal features. Broadly, acoustic features can be grouped into two categories: time-domain (or temporal features) and frequency-domain (or spectral features). A number of those have been proposed in the literature.

Two widely used time-domain measures are given as follows. [22].

- Short-time energy:

$$E_n = \frac{1}{N}\sum_m [x(m)w(n-m)]^2$$

where $x(m)$ is the discrete time audio signal, n is the time index of the short-time energy, and $w(m)$ is the window of length N. Short-time energy provides a convenient representation of the amplitude variation over time.

- Short-time average zero-crossing rate (ZCR):

$$Z_n = \frac{1}{2}\sum_m |\mathrm{sgn}[x(m)] - \mathrm{sgn}[x(m-1)]|w(n-m)$$

where

$$\mathrm{sgn}[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) < 0. \end{cases}$$

Zero-crossings occur when successive samples have different signs, and the ZCR rate is the average number of times the signal changes its sign within the short-time window. We calculate both energy and ZCR values using a window of 256 samples with a 50% overlap, at an input sampling rate of 22 050 Hz.

Similarly, a variety of spectral features have been proposed. These features are typically obtained by first applying a Fourier transform [implemented as a fast Fourier transform (FFT)] to short-time window segments of audio signals followed by further processing to derive the features of interest. Some commonly used ones include the following.

- *MFCC* [29]: After taking the FFT of each short-time window, the first step in MFCC calculation is to obtain the mel-filter bank outputs by mapping the powers of the spectrum onto the mel scale, using 23 triangular mel-filterbanks, and transformed into a logarithmic scale, which emphasizes the low varying frequency characteristics of the signal. Typically 13 mel frequency cepstral coefficients are then obtained by taking the discrete cosine transform (DCT).
- *Band Energy Ratio* [11]: It is the ratio of the energy in a specific frequency-band to the total energy. Eight logarithmic sub-bands are used in our experiments.
- *Spectral Flux* [19]: It is used to measure a spectral amplitude difference between two successive frames.
- *Statistical Moments* [19], [30]: The commonly used statistical moments include the following.
  - *Spectral Centroid* measures the brightness of a sound. The higher the centroid, the brighter the sound.
  - *Signal Bandwidth* measures the width of the range of signal's frequencies.
  - *Spectral Flatness* quantifies the tonal quality; namely, how much tone-like the sound is as opposed to being noise-like.
  - *Spectral Roll-Off* quantifies the frequency value at which the accumulative value of the frequency response magnitude reaches a certain percentage of the total magnitude. A commonly used threshold is 95%.

Another commonly used feature is linear prediction cepstral coefficients (LPCCs) [31]. The basic idea behind linear prediction is that the current sample can be predicted, or approximated, as a linear combination of the previous samples, which would provide a more robust feature against sudden changes. LPCC is calculated using the autocorrelation method in this work [29].

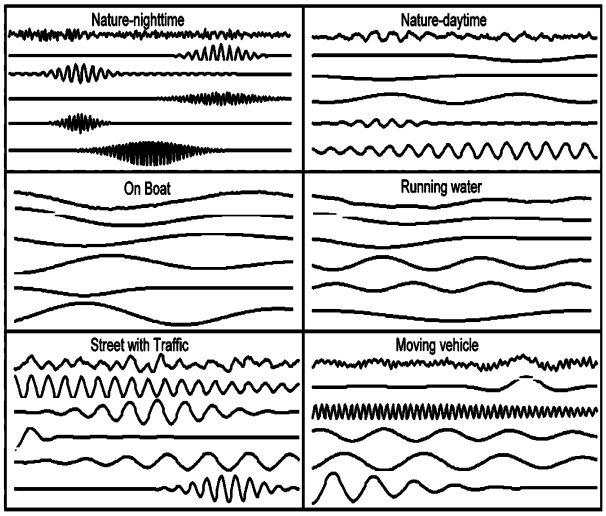

Fig. 1. Illustration of the decomposition of signals from six different classes as listed, where the top-most signal is the original, followed by the first five basis vectors.

Most previous efforts utilize a combination of some, or even all, of the aforementioned features, to characterize audio signals. However, adding more features is not always helpful. As the feature dimension increases, data points become sparser and there are potentially irrelevant features that could negatively impact the classification result. We showed in [5] that the use of all features for classification does not always produce good performance for the audio classification problems of our interest. This in turn leads to the issue of selecting an optimal subset of features from a larger set of possible features to yield the most effective subset. In [5], we utilized a simple feature selection algorithm to obtain a smaller feature set to reduce the computational cost and running time and achieve an acceptable, if not higher, classification rate. Although the results showed improvements, the features found after the feature selection process were found to be specific to each classifier and environment type. A similar phenomenon was observed in [13], where different feature subsets were tried to increase the performance for each context type. It was with these findings that motivated us to look for a more effective and principled approach for determining an appropriate representation for environmental sound classification. Toward this goal, we propose the use of MP as a new feature selection method.

B. Signal Representation With Matching Pursuit (MP)

The intuition behind our strategy is that there are underlying structures that lie within signals of each type of environment, and we could use MP to discover them. Different types of environmental sounds have their own unique characteristics, making the decomposition into sets of basis vectors to be noticeably different from one another. By using a dictionary that consists of a wide variety of functions, MP provides an efficient way of selecting a small set of basis vectors that produces meaningful features as well as flexible representation for characterizing an audio environment. Examples of the decompositions of signals from six sound classes using Gabor atoms, described in Section IV-B, is shown in Fig. 1, where the top five atoms are shown.

To achieve an efficient representation, we would like to obtain the minimum number of basis vectors to represent a signal, resulting in a sparse approximation. However, this is an NP-complete problem. Various adaptive approximation techniques to obtain such a signal representation in an efficient manner have been proposed in the literature, including basis pursuit (BP) [32], matching pursuit (MP) [33], and orthogonal matching pursuits (OMP) [34]. All of these methods utilize the notion of a dictionary that capacitates the decomposition of a signal by selecting basis vectors from a given dictionary to find the best subset.

BP provides a framework that minimizes the L1-norm of coefficients occurring in the representation, but at a cost in linear programming. Although it provides good representations, BP is computationally intensive. By using a dictionary that consists of a wide variety of elementary waveforms, MP aims at finding sparse decompositions of a signal efficiently in a greedy manner. MP is suboptimal in the sense that it may not achieve the sparsest solution. Usually, elements in a given dictionary are selected by maximizing the energy removed from the residual signal at each step. Even in just a few steps, the algorithm can yield a reasonable approximation with a few atoms, and the decomposition will provide us with an interpretation of the signal structure. We adopt the classic MP approach to generate audio features in our study.

The MP algorithm was originally introduced by Mallat and Zhang [33] for decomposing signals in an overcomplete dictionary of functions, providing a sparse linear expansion of waveforms. As long as the dictionary is overcomplete, the expansion is guaranteed to converge to a solution where the residual signal has zero energy. The following description of the MP algorithm is based on the descriptions from [32].

Let dictionary D be a collection of parameterized waveforms ϕ_γ given by

$$D = \{\phi_\gamma : \gamma \in \Gamma\}$$

where Γ is the parameter set and ϕ_γ is called an atom. The approximate decomposition of a signal s can be written as

$$s = \sum_{i=1}^{m} \alpha_{\gamma_i} \phi_{\gamma_i} + R^{(m)} \tag{1}$$

where $R^{(m)}$ is the residual. Given s, m, and D, our goal is to find indices γ_i and compute α_{γ_i}, where $i = 1, 2, \ldots, m$, while minimizing $R^{(m)}$. Starting from initial approximation $s^{(0)} = 0$ and residual $R^{(0)} = s$, the MP algorithm builds up a sequence of sparse approximation stepwise.

Initially, the MP algorithm computes all inner products of signal s with atoms in dictionary D. The atom with the largest magnitude inner product ϕ_{γ_0} is selected as the first element. Thus, the atom selection criteria can be given as

$$|\langle s, \phi_{\gamma_0} \rangle| \geq |\langle s, \phi_\gamma \rangle| \quad \forall\, \gamma \in \Gamma.$$

After the first step, atom ϕ_{γ_0} is subtracted from s to yield residual $R^{(0)}$. Generally, at stage $k = 1, 2, \cdots$, the MP algorithm identifies the atom that best correlates with the residual

and then adds the scalar multiple of that atom to the current approximation

$$s^{(k)} = s^{(k-1)} + \alpha_k \phi_{\gamma_k} \tag{2}$$

where

$$\alpha_k = \left\langle R^{(k-1)}, \phi_{\gamma_k} \right\rangle \quad \text{and} \quad R^{(k)} = s - s^{(k)}.$$

After m steps, one has a representation of the approximate decomposition with residual $R = R^{(m)}$ as shown in (1).

Various dictionaries have been proposed to be used with MP, including wavelets [35], wavelet packets [36], cosine packets [37], Gabor dictionaries [33], multiscale Gabor dictionaries [37], [38], Chirplets [39], and others. Most dictionaries are complete or overcomplete, and the approximation techniques, such as MP, allow for the combination of different dictionaries. Examples of some basic dictionaries are: 1) frequency (i.e., Fourier functions), 2) time-scaled (i.e., Haar wavelets), and 3) time–frequency, (i.e., Gabor functions). To encapsulate the nonstationary characteristics of audio signals, we use a dictionary of Gabor atoms to offer a more discriminant time–frequency representation. In Section IV-B, we will discuss this in further detail.

IV. Feature Extraction With Matching Pursuit (MP)

Desirable types of features should be robust, stable, and straightforward, with the representation being sparse and physically interpretable. We will show that using MP will make this representation possible. The advantages of this representation are the ability to capture the inherent structure within each type of signal and to map from a large, complex signal onto a small, simple feature space. More importantly, it is conceivably more invariant to background noise and could capture characteristics in the signal where MFCCs tend to fail. In this section, we will describe how MP features are obtained.

A. Extracting MP Features

Our goal is to use MP as a tool for feature extraction for classification, and not necessarily to recover or approximate the original signal for compression. Nevertheless, MP provides an excellent way to accomplish either of these tasks. MP is a desirable method to provide a coarse representation and to reduce the residual energy with as few atoms as possible. The decomposition from MP also furnishes us with an interpretation of the signal structures. The strategy for feature extraction is based on the assumption that the most important information of a signal lies in leading synthesizing atoms with the highest energy, yielding a simple representation of the underlying structure. Since MP selects atoms in order by eliminating the largest residual energy, it lends itself in providing the most useful atoms, even just after a few iterations.

The MP algorithm selects atoms in a stepwise manner among the set of waveforms in the dictionary that best correlate the signal structures. The iteration can be stopped when the coefficient associated with the atom selection falls below a threshold or when a certain number of atoms selected overall has been reached. Another common stopping criterion is to use the signal to residual energy ratio. In this paper, we chose n atoms as the stopping criterion for the iteration. MP features are selected by the following process.

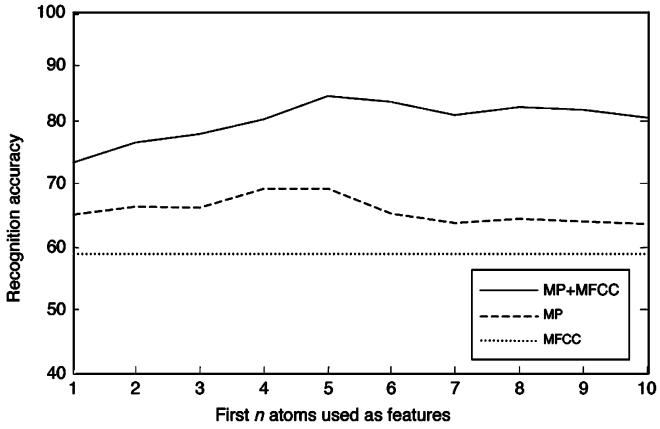

Fig. 2. Comparison of classification rates (with the GMM classifier) using the first n atoms, where $n = 1, \ldots, 10$, as features while the MFCC features are kept the same.

Based on our experimental setup, explained in Section V-A, we use a rectangular window of 256 points with a 50% overlap. This corresponds to the window size used for all feature extraction. We decompose each 256-point segment using MP with a dictionary of Gabor atoms that are also 256 points in length. We stop the MP process after obtaining n atoms. Afterwards, we record the frequency and scale parameters for each of these n atoms and find the mean and the standard deviation corresponding to each parameter separately, resulting in four feature values.

To select parameter n in the stopping criterion, we plot the classification performance as a function of n in Fig. 2. It shows a rise with an increasing number of features due to the increased discriminatory power with the performance leveling off around four or five atoms. Thus, we chose $n = 5$ atoms in our experiments and use the same process to extract features for both training and test data. The decomposition of different signals from the same environmental class might not be composed of exactly the same atoms or order. However, since we are taking the average of their parameters as features, the sequencing order of atoms is neglected and the robustness of these features is enhanced by averaging. Using these atom parameters as features abstracts away finer details and forces the concentration on the most pronounced characteristics.

The above truncation process is similar to that of non-injective mapping. When mapping a large problem space into the feature space, only a few significant features are considered, enabling us to disregard the rest. The most important information in describing a signal could be found in a few basis vectors with the highest energies, and the process in which MP selects these vectors are exactly in the order of eliminating the largest residual energy. This means that even the first few atoms found by MP will naturally contain the most information, making them to be more significant features. This also allows us to map each signal from a larger problem space into a point in a smaller feature space. Any data items are similar as long as their representation in the feature space are similar or close in proximity.

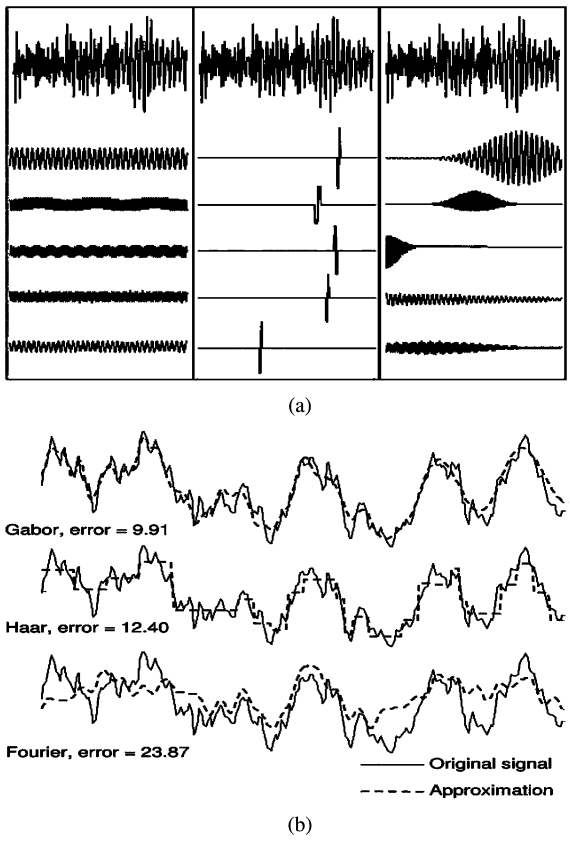

Fig. 3. (a) Decomposition of signals using MP (the first five basis vectors) with dictionaries of Fourier (left), Haar (middle), and Gabor (right), and (b) approximation (reconstruction) using the first ten coefficients from MP with dictionaries of Gabor(top), Haar (middle), and Fourier (bottom).

B. MP Dictionary Selection

Examples of the MP decomposition using different dictionaries are compared in Fig. 3. The first five atoms obtained from the MP decomposition with Fourier, Haar, and Gabor dictionaries are shown in Fig. 3(a). Since the Fourier representation is formed by the superposition of non-local signals, it demands a large number of atoms for cancellation to result in a local waveform. In contrast, the Gabor representation is formed by a band-limited signal of finite duration, thus making it more suitable for time–frequency localized signals. The Gabor representation was shown in [40] to be optimal in the sense of minimizing the joint two-dimensional uncertainty in the combined spatial-frequency space. The effectiveness of reconstructing a signal using only a small number of atoms is compared in Fig. 3(b), where ten atoms are used. Gabor atoms result in the lowest reconstruction error, as compared with the Haar or the Fourier transforms using the same number of coefficients. Due to the nonhomogeneous nature of environmental sounds, using features with these Gabor properties would benefit a classification system. Based on the above observation, we choose to use the Gabor function in this work.

Gabor functions are sine-modulated Gaussian functions that are scaled and translated, providing joint time–frequency localization. Mathematically, the discrete Gabor time–frequency atom is written as

$$g_{s,u,\omega,\theta}(n) = \frac{K_{s,u,\omega,\theta}}{\sqrt{s}} e^{-\pi(n-u)^2/s^2} \cos[2\pi\omega(n-u)+\theta]$$

where $s \in \mathbb{R}^+; u, \omega \in \mathbb{R}; \theta \in [0, 2\pi]$. $K_{s,u,\omega,\theta}$ is a normalization factor such that $\|g_{s,u,\omega,\theta}\|^2 = 1$. We use $\gamma = (s, u, \omega, \theta)$ to denote parameters of the Gabor function, where s, u, ω, and θ correspond to an atom's position in scale, time, frequency, and phase, respectively. The Gabor dictionary in [33] was implemented with atom parameters chosen from dyadic sequences of integers. The scale s, which corresponds to the atom width in time, is derived from dyadic sequence $s = 2^p$, $1 \le p \le m$, and the atom size is equal to $N = 2^m$.

We chose the Gabor function with the following parameters in this work, $s = 2^p (1 \le p \le 8), u = \{0, 64, 128, 192\}$, $\omega = Ki^{2.6}$ (with $1 \le i \le 35$, $K = 0.5 \times 35^{-2.6}$ so that the range of ω is normalized between 0 and 0.5), $\theta = 0$ and the atom length is truncated to $N = 256$. Thus, the dictionary consists of $1120 = 8 \times 35 \times 4$ Gabor atoms that were generated using scales of 2^p and translation by quarters of atom length N.

We attempt to keep the dictionary size small since a large dictionary demands higher complexity. For example, we choose a fixed phase term since its variation does not help much.

By shifting the phase, i.e., $\theta = \{0, (\pi)/(4), (\pi)/(2), \ldots\}$, each basis vector only varies slightly. Since we are using the top few atoms for creating the MP-features, it was found not necessary to incorporate the phase-shifted basis vectors.

A logarithmic frequency scale is used to permit a higher resolution in the lower frequency region and a lower resolution in the higher frequency region. We found the exponent 2.6 in ω experimentally given the parameter setting of the frequency interval. We wanted to have a finer granularity below 1000 Hz as well as enough descriptive power in the higher frequency. The reason for finer granularity in lower frequencies is because more audio object types occur in this range, and we want to capture finer differences between them.

We can observe differences in synthesizing atoms for different environments, which demonstrates that different environments exhibit different characteristics, and each set of decompositions encapsulates the inherent structures within each type of signal. For example, because the two classes, *On boat* and *Harbor*, contain ocean sounds, the decompositions are very similar to each other. Another example is between *Nature-daytime* and *Near highway*. Both were recorded outdoors; therefore, there are some similarities in the subset of their decomposition but because the *Near highway* class has the presence of traffic noise, this has led to distinctively different atoms with higher frequency components, compared to *Nature-daytime*. When we compared them with differing classes, e.g., *Nature-nighttime* and *Near highway*, the decompositions are noticeably different from one another. Therefore, we utilize these set of atoms as a simple representation to these structures.

C. Computational Cost of MP Features

For each input audio signal, we divide into k overlapping windows of length N, and MP is performed on each of these

k windows. At each iteration, the MP algorithm computes the inner product of the window of signals (or residuals) with all D atoms in the dictionary. The cost of computing all inner products would be $O(kND)$. During this process, we need to record the highest correlation value and the corresponding atom. We terminate the MP algorithm after n iterations, yielding a total cost of $O(nkND)$. By keeping the dictionary size small with constant iteration number n and window size N, the computational cost is a linear function of the total length of the signal. Thus, it can be done in real time.

V. Experimental Evaluation

A. Experimental Setup

We investigated the performance of a variety of audio features and provide an empirical evaluation on 14 different types of environmental sounds commonly encountered. We used recordings of natural (unsynthesized) sound clips obtained from [41] and [42]. We used recordings that are available in WAV formats to avoid introducing artifacts in our data (e.g., from the MP3 format). Our auditory environment types were chosen so that they are made up of nonspeech and nonmusic sounds. It was essentially background noise of a particular environment, composed of many sound events. We do not consider each constituent sound event individually, but as many properties of each environment. Naturally, there could be infinitely many possible combinations. To simplify the problem, we restricted the number of environment types examined and enforced each type of sound to be distinctively different from one another, which minimized overlaps as much as possible. The fourteen environment types considered were: *Inside restaurants, Playground, Street with traffic and pedestrians, Train passing, Inside moving vehicles, Inside casinos, Street with police car siren, Street with ambulance siren, Nature-daytime, Nature-nighttime, Ocean waves, Running water/stream/river, Raining/shower*, and *Thundering*.

We examined the performance of the MP features, extracted as described in Section IV, a concatenation of the MP-features and MFCCs to form a longer feature vector, MP+MFCC (16), and a variety of commonly used features, which includes MFCC (12), ΔMFCC (12), LPC (12), ΔLPC (12), LPCC(12), the band energy ratio, frequency roll-off set at 95%, spectral centroid, spectral bandwidth, spectral asymmetry, spectral flatness, zero-crossing, and energy. We adopted the GMM classification method in the feature space for our work. With GMMs, each data class was modeled as a mixture of several Gaussian clusters. Each mixture component is a Gaussian represented by the mean and the covariance matrix of the data. Once the model was generated, conditional probabilities were computed using

$$p(x|X_k) = \sum_{j=1}^{m_k} p(x \mid j)P(j)$$

where X_k is the datapoints for each class, m_k is the number of components, $P(j)$ is the prior probability that datum x was generated by component j, and $p(x|j)$ is the mixture component density. The EM algorithm [43] was then used to find the maximum likelihood parameters of each class.

We also investigated the K-nearest neighbor (kNN) classification method. kNN is a simple supervised learning algorithm where a new query is classified based on the majority class of its k nearest neighbors. A commonly used distance measure is the Euclidean distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

In our experiments, we utilized separate source files for training and test sets. We kept the 4-s segments that were originated from the same source file separate from one another. Each source file for each environment was obtained at different locations. For instance, the *Street with traffic* class contains four source files which were labeled as taken from various cities. We required that each environment contained at least four separate source recordings, and segments from the same source file were considered a set. We used three sets for training and one set for testing. Finally, we performed a fourfold cross validation for the MP features and all commonly used features individually for performance comparison. In this setup, none of the training and test items originated from the same source. Since the recordings were taken from a wide variety of locations, the ambient sound might have a very high variance. Results were averaged over 100 trials. These sound clips were of varying lengths (1–3 min long), and were later processed by dividing up into 4-s segments and downsampled to 22 050 Hz sampling rate, mono-channel and 16 bits per sample. Each 4-s segment makes up an instance for training/testing. Features were calculated from a rectangular window of 256 points (11.6 ms with 50% overlap.

B. Experimental Results

We compare the overall recognition accuracy using MP, MFCC, and their combination for 14 classes of sounds in Fig. 4. As shown in this figure, MFCC features tend to operate on the extremes. They perform better than MP features in six of the examined classes while producing extremely poor results in the case of five other classes; namely, a recognition rate of 0% for four classes, *Casino, Nature-nighttime, Train passing*, and *Street with ambulance* and less than 10% for *Thundering*. MP features perform better overall, with the exception of two classes (*Restaurant* and *Thundering*) having the lowest recognition rate at 35%. One illustrative example is the *Nature-nighttime* class, which contains many insect sounds of higher frequencies. Unlike MFCCs that recognized 0% of this category, MP features were able to yield a correct recognition rate of 100%. Some of these sounds are best characterized by narrow spectral peaks, like chirps of insects. MFCC is unable to encode such narrow-band structure, but MP features are effective in doing so. By combining MP and MFCC features, we were able to achieve an averaged accuracy rate of 83.9% in discriminating fourteen classes. There are seven classes that have a classification rate higher than 90%. We see that MFCC

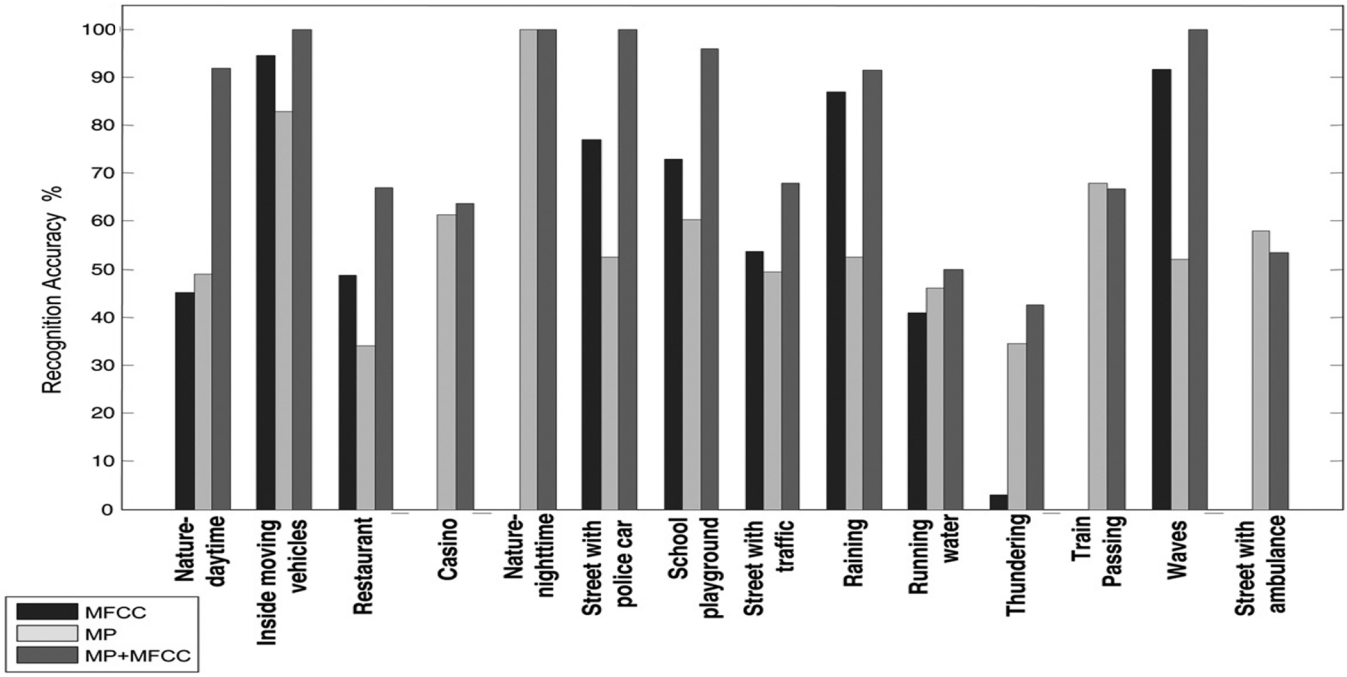

Fig. 4. Overall recognition rate (GMM) comparing 14 classes using MFCC only, MP only, and MP+MFCC as features. (0% recognition for four classes using MFCC only: *Casino,Nature-nighttime,Train passing*, and *Street with ambulance*).

and MP features complement each other to give the best overall performance.

For completeness, we compared the results from the two different classifiers, namely GMM and kNN. We examine the results from varying the number of neighbors K and using the same K for each environment type. The overall recognition rate by varying K are given in Fig. 5. The highest recognition rate was obtained using $K = 32$, with an accuracy of 77.3%. We could observe the performance slowly flattens out and further degrades as we increase the number of neighbors. By increasing K, we are in fact expanding the radius of its neighbors. Extending this space makes it more likely the classes would overlap. In general, the results from GMM outperforms those from using kNN. Therefore, we will concentrate on GMM for the rest of our experiments. Using GMM allows for better generalization. kNN would perform well if the data samples are very similar to each other. However, since we are using different sources for testing and training, they might be similar in their overall structure but not finer details.

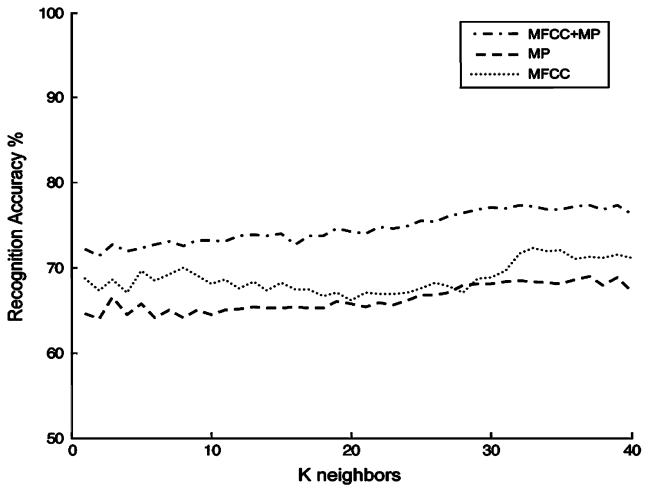

Fig. 5. Overall recognition accuracy using kNN with varying number of K.

To determine the model order of GMM, we examine the results by varying the number of mixtures. Using the same settings as the rest of the experiments, we examined mixtures of 1–10, 15, and 20 and used the same number of mixtures for each environment type. The overall recognition rates are given in Table I. We see that the classification performance peaks around five mixtures and the performance slowly degrades as the number of mixtures increases. The highest recognition rate for each class across the number of mixtures was obtained with 4–6 mixtures. They were equal to 4, 5, 5, 5, 5, 5, 6, 5, 4, 4, 5, 6, 5, 5 for the corresponding classes: *Nature-daytime, Inside moving vehicles, Inside restaurants, Inside casinos, Nature-nighttime, Street with police car siren, Playground, Street with traffic, Thundering, Train passing, Raining/shower, Running water/stream, Ocean waves*, and *Street with ambulance*. We also experimented with this combination of mixtures numbers, and the results is given as *mixed* in Table I. Since the latter requires tailoring to each class, we decided to just use five mixtures throughout all of our experiments to avoid making the classifier too specialized to the data. We performed an analysis of variance (ANOVA) on the classification results. Specifically, we used the t-test, which is a special case of ANOVA for comparing two groups. The t-test was run on each of the 14 classes individually. The t-tests showed that the result of the two systems was significant with $p < 0.001$ for all 14 classes.

An interesting benchmark is shown in Fig. 6, where we ran the same experiments using all features, including MP, MFCC, and other commonly used features as stated in Section V-A. The

TABLE I
RECOGNITION ACCURACY USING GMM WITH A VARYING NUMBER OF MIXTURES, USING MFCC AND MP FEATURES

Number of mixtures	1	2	3	4	5	6	7	8	9	10	15	20	mixed
Accuracy (%)	68.1	66.8	74.6	76.0	83.9	80.4	77.5	73.9	73.2	73.7	70.8	69.4	83.4

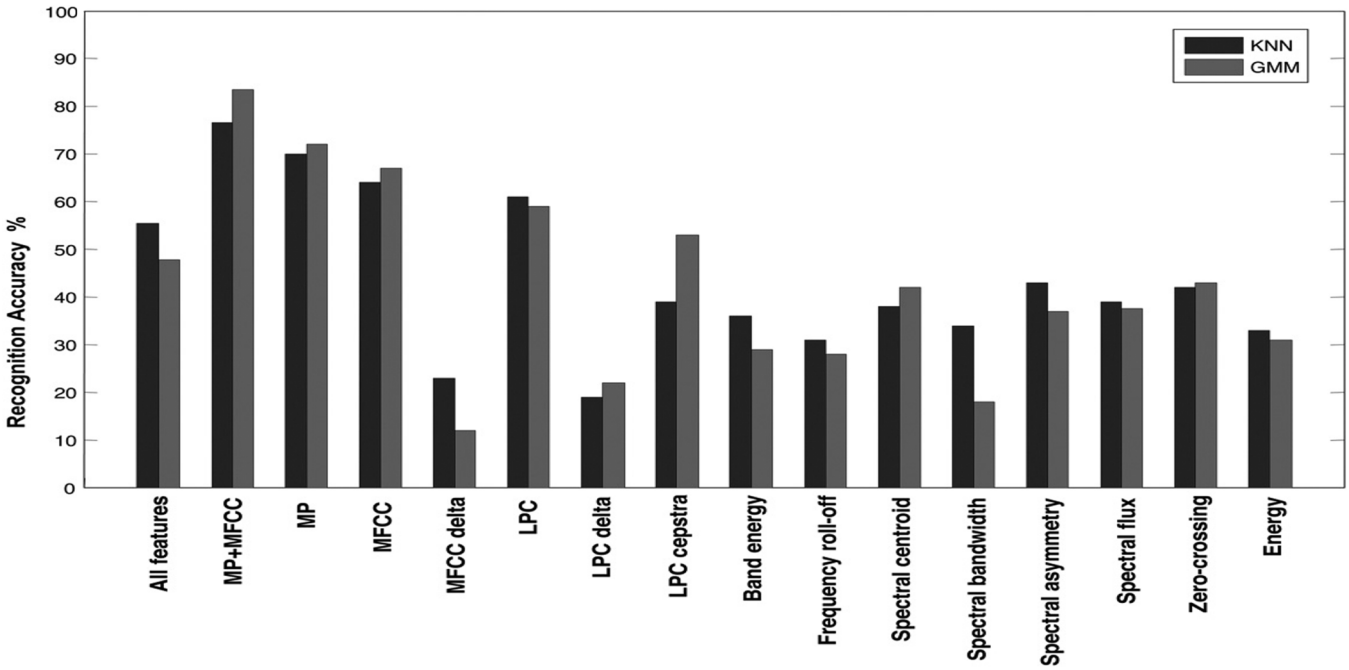

Fig. 6. Overall recognition accuracy comparing MP, MFCC, and other commonly used features for 14 classes of sounds using kNN and GMM as classifiers.

TABLE II
CONFUSION MATRIX FOR 14-CLASS CLASSIFICATION USING MP FEATURES AND MFCC WITH GMM

	Nature-daytime	Inside vehicle	Restau-rant	Casino	Nature-nighttime	Street - police	Play-ground	Street - traffic	Thun-dering	Train	Rain / shower	Stream/ river	Waves	Street - ambulance
Nature-daytime	92.2													
Vehicle		100												
Restaurant			66.8				2.5	12.8	8.6					
Casino			23.0	62.2										
Nature-nighttime					100									
Police			1.8	33.7		97.5								4.4
Playground			2.9				94.6					13.5		
Traffic			1.2					74.8		5.9				3.5
Thundering						1.3	12.1	91.1	11.2		7.1			
Train										60.7				
Rain											46.5			
Stream			3.8	3.5			1.2				53.3	78.3		37.1
Waves										22.0			100.0	
Ambulance			2.2											54.6

average recognition accuracy is approximately 55.2%, which is much worse than using combined MFCC and MP features. This confirms our discussion in Section III-A; namely, adding more features may not be always helpful.

C. Confusion Matrix and Pairwise Classification

Results presented in Section V-B are averaged values from all trials together. To further understand the classification performance, we show results in the form of a confusion matrix, which allows us to observe the degree of confusion among different classes. The confusion matrix given in Table II is built from a single arbitrary trial, constructed by applying the classifier to the test set and displaying the number of correctly/incorrectly classified items. The rows of the matrix denote the environment classes we attempt to classify, and the columns depict classified results. We see from Table II that *Restaurant, Casino, Train, Rain*, and *Street ambulance* were more often misclassified than the rest. We could further point out that the misclassification overlaps between pairs, such as those of *Inside restaurant*

TABLE III
RECOGNITION ACCURACY FOR PAIRWISE CLASSIFICATION USING GMM

	Nature-daytime	In moving vehicle	Restaurant	Casino	Nature-night-time	Street w/ police car siren	Playground	Street w/ traffic	Thundering	Train passing	Rain/ shower	Stream/ running water	Waves	Street w/ ambulance
Nature-daytime														
In moving vehicle														
Restaurant														
Casino			58.1											
Nature-night-time														
Street w/ police car														
Playground														
Street			85.6											
Thundering				86.2										
Train passing								81.8						
Rain/ shower							61.2							
Stream/ Running water							85.3				75.3			
Waves										84.1				
Street w/ ambulance														

TABLE IV
COMPARISON OF RECOGNITION ACCURACY BETWEEN MFCC, MP, AND MFCC+MP FEATURES FOR PAIRWISE CLASSIFICATION OF FIVE-CLASS EXAMPLES. FOR EACH PAIR OF CLASSES, THE THREE RECOGNITION ACCURACY VALUES CORRESPOND TO: (LEFT) MFCC, (MIDDLE) MP, (RIGHT) MFCC+MP FEATURES. ALL VALUES ARE IN PERCENTAGES

	Nature-daytime			Nature-night-time			Playground			Rain / shower		
Nature-night-time	47.6	50	89.7									
Playground	50	86.4	99.5	66.5	50	99.3						
Rain / shower	50	50	98.4	50	100	100	50	61.2	61.2			
Stream / running water	50	100	100	100	98.3	98.9	46.2	67.4	85.3	58.1	60.5	75.3

and *Inside casino* and of *Rain* and *Steam (Running River)*. Interestingly, there exists a one-sided confusion between *Train* and *Waves*, where samples of *Train* were misclassified as *Waves*, but not vice versa.

Generating a confusion matrix provides a convenient way to understand the performance of features and classifiers. However, since it is obtained from all classes, it is difficult to observe more subtle details. In many instances, we are interested in determining where misclassification actually occurs; namely, whether it is originating from the classifier or the ambiguity of extracted features. To address this, we use a pairwise classification method to observe the interaction between all possible pairs of classes. Pairwise classification is a series of two-class problems in a one-against-one manner, instead of the one-against-all method used to construct the confusion matrix. By examining all exhaustive pairs of classes and finding the most difficult ones, we show the pairwise classification results in Table III. For most pairs of classes, we obtained a correct classification rate higher than 90%. Only cases with correct classification rates less than 90% are listed in Table III. A simple two-class classification result is around 58% in differentiating classes between inside *restaurant* or *casino*, which is not much better than random guessing.

We investigate more closely the effectiveness of MP features by presenting the pairwise classification results for five classes of environmental sounds, with 20 data samples each. By examining a smaller problem, we could observe the subtle details of their classification performance. The five classes are *Playground, Nature-daytime, Nature-nighttime, Stream/river*, and *Raining*. Table IV shows ten pairwise classification results between five classes. For each pair of classes, recognition rates are given in three boxes. They correspond to the use of different features for classification: MFCC features only (left), MP features only (middle) and joint MFCC and MP features (right). The use of joint MFCC and MP features tends to result in a higher accuracy rate. One impressive example is observed in discriminating *Rain/shower* and *Nature-daytime*, the use of MFCC and MP-features alone results in only an accuracy rate

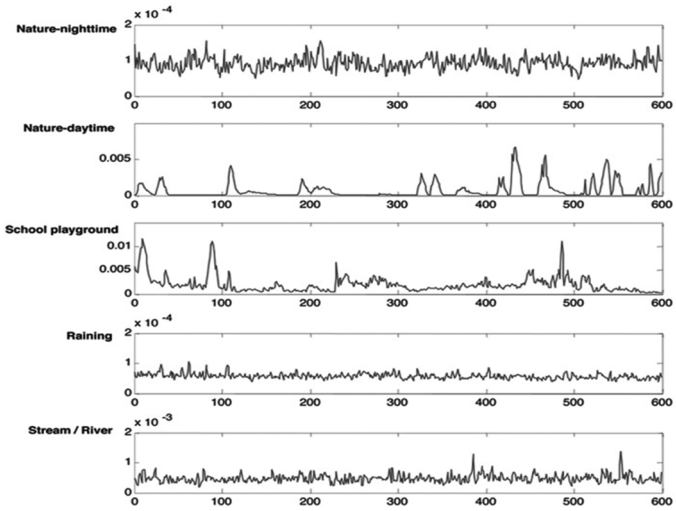

Fig. 7. Sample of the short-time energy function from each of the example five classes. (a) Nature-nighttime. (b) Nature-daytime. (c) Playground. (d) Raining. (e) River/stream.

of 50%. However, the use of two types of features jointly leads to an accurate classification rate of 98.4%.

D. Comparison of Time-Domain Features

Some environment sounds may include strong temporal domain signatures such as those from chirpings of insects and raining, which are noise-like with a broad flat spectrum. These characteristics might be better captured with temporal type features. When compared with spectral features, there are fewer temporal-domain features used to characterize audio signals. Two commonly used temporal features are the short-time energy and the zero-crossing rate [22]. In this paper, we present new temporal features based on MP. In this subsection, we would like to compare these three features.

Fig. 7 provides an example of the short-time energy function of signals from five different classes. However, it may not provide an effective discriminant feature as illustrated in Fig. 8(a), where we show the energy range of twenty data samples for five sound classes. We see from Fig. 8(a) that the energy range of *Nature-nighttime* resembles a flat line. This is due to the high frequency in the chirping of insects, making it similar to a constant sound. The large variation within each type of sounds also makes it difficult to determine the effectiveness of each feature for each sound type. The zero-crossing rate can be useful to separating some classes such as *Nature-nighttime* and *Raining* from the rest of the classes as shown in Fig. 8(b). However, the other three types have very similar properties and, thus, they are more difficult to distinguish.

MP features provide a more flexible and effective way to extract temporal features of environmental sounds using time- and frequency-localized representation. For illustration, the mean distribution of three types of MP parameters are shown in Fig. 9. We see that these MP features form clearly separable clusters among themselves. For example, the *Nature-nighttime* class makes a cluster in the higher frequency and smaller scales due to the fact that insects have high-pitched repeating chirps. In contrast, running streams of water produce a lower frequency

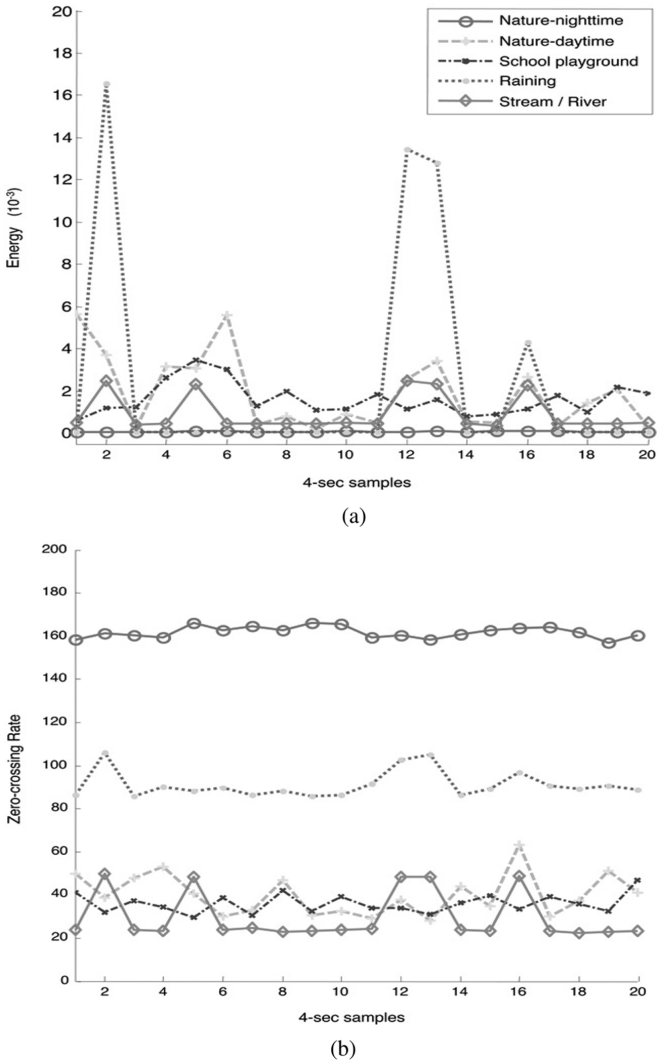

Fig. 8. Temporal features: (a) the energy range and (b) the zero-crossing rate. (Figures (a) and (b) share the same legend.)

sound, and they are mapped to the lower frequency and higher scale region in the figure.

Using similar experiment settings as in Section V-A, we perform classification on these five classes using GMM. We obtained results of 75.3%, 84.0%, and 89.7% for MFCC only, MP-features only, and the combined MFCC+MP-features, respectively. Similar to previous findings, including MP-features with MFCCs in the feature vector increases classification performance than using MFCCs alone. To achieve a better understanding of how combining MFCCs with individual MP-feature descriptors helps with classification, we can observe the results in first row of the Table V, where we perform the classification using the input feature vector as a combination, or more specifically concatenation, of MFCCs with one (or two) of the descriptors at a time. We use *mean-F* and *std-F* to denote the mean and standard deviation for the frequency indices and likewise, *mean-S* and *std-S* for the scale indices. Table V shows how the descriptors contributes to the overall classification. We further observe how each descriptor affects certain classes by repeating the experiment with pairwise classifications as listed in Table V. We see that the effect of each descriptor is different for each pair

TABLE V
COMPARISON OF RECOGNITION ACCURACY BETWEEN MFCC AND MFCC WITH INDIVIDUAL MP FEATURES FOR PAIRWISE AND OVERALL CLASSIFICATION OF THE FIVE-CLASS EXAMPLES USING GMM, IN PERCENTAGE

	MFCC	+ MP-features	+ mean-F	+ std-F	+ mean-, std-F	+ mean-S	+ std-S	+ mean-, std-S
5-class (Overall)	65.3	86.7	80	78.7	82.7	73.3	69.3	78.7
Nature-daytime - Nature-nighttime	47.6	89.7	75.3	75.3	61.2	85.3	66.5	66.5
Nature-daytime - Playground	50	99.5	66.5	75.3	75.3	50	89.7	66.5
Nature-daytime - Rain	50	98.4	61.2	77.3	61.2	75.3	75.3	85.3
Nature-daytime - Stream	50	100	100	100	100	85.3	85.3	85.3
Nature-nighttime - Playground	66.5	99.3	100	75.3	100	75.3	85.3	75.3
Nature-nighttime - Rain	50	100	100	85.3	100	89.7	85.3	85.3
Nature-nighttime - Stream	100	98.9	100	98.9	100	100	98.9	98.9
Playground - Rain	50	61.2	75.3	66.5	75.3	47.6	50	50
Playground - Stream	46.2	85.3	66.5	61.2	61.2	61.2	50	66.5
Rain - Stream	58.1	75.3	75.3	75.3	75.3	66.5	61.2	61.2
Nature-nighttime - Thundering	0	100	100	0	100	0	0	0

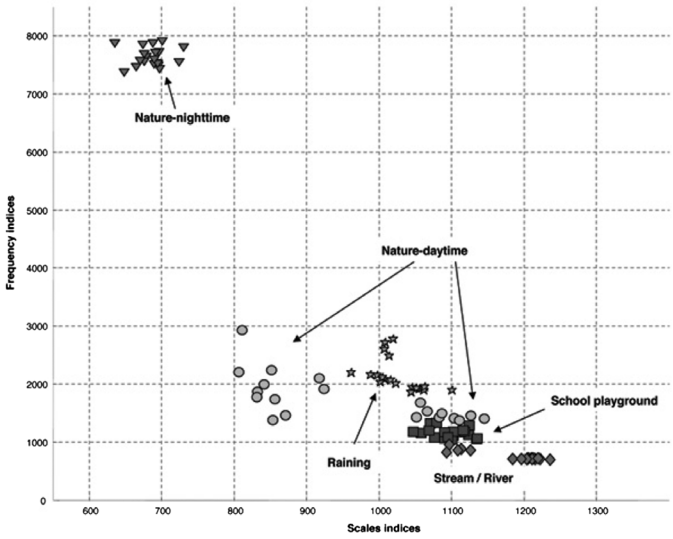

Fig. 9. MP features (i.e., the mean value of the corresponding parameters) in feature space.

of environments. To further examine the effects, we plotted the values to each of the descriptor in Fig. 10.

The *mean-S* can be viewed as an indication of the overall amplitude of the signal. It depends on the loudness of the signal or how far away the microphone is from the sound source. The *std-S* descriptor provides us with a way to disclose the variability of the energy in the time–frequency plane. The values for static type of noises, such as those of constant raining, are higher than diverse noises. Another interesting observation is that out of the four descriptors, *std-S* was the only one that separates out much of the *Nature-daytime* class from the others, which was the most difficult to do with the other descriptors. The *mean-F* might be similar to that of the centroid as it represents where the energy on the frequency axis. Although, the mean-F only describes the frequency, but it still proved to be useful when combined with MFCC. One of the reason is that MFCCs model the human auditory system and do poorly when modeling nonspeech type noise. *Mean-F* furnishes us with a description of the basic frequency without being modeled based on any auditory system. *Std-F* expresses the frequency range. If the sound frequency is narrow, *std-F* is low, i.e., running stream. An interesting example is for the class, between *Nature-Nighttime* and *Thundering*, where using MFCCs alone yields 0%. However, we can see in Table V that adding the *mean-F* to the feature vector helps significantly. In this case, *mean-S* was less im-

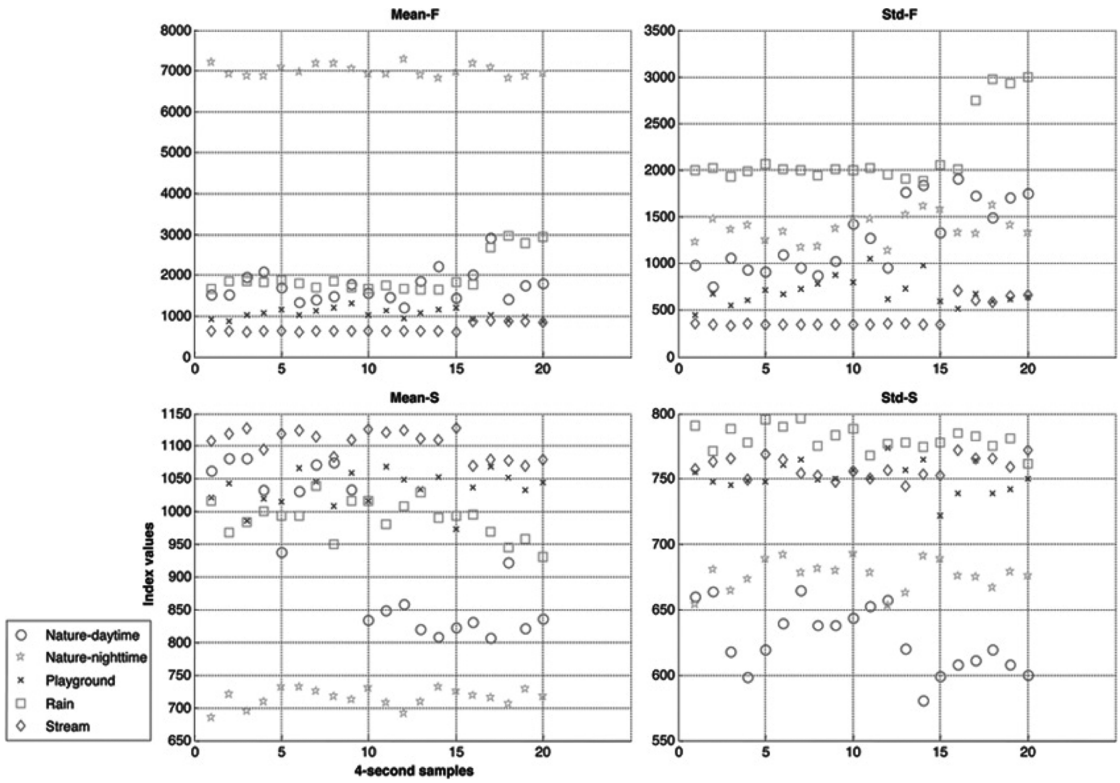

Fig. 10. Individual MP feature descriptor values: mean-F (top left), std-F (top right), mean-S (bottom left), std-S (bottom right).

portant in discriminating between *Nature-Nighttime* and *Thundering*, which also indicates that it is not relying on the amplitude of the signal. We can see that although different descriptors might be better for certain pair of classes, it would be difficult, and too specific, to selectively choose them, but from Table V, we can conclude that using all the frequency and scale descriptors provides us with extra information for discriminating between difficult classes.

VI. Listening Tests

A. Test Setup and Procedure

A listening test was conducted to study human recognition capability of these environmental sounds. Our motivation was to find another human-centric performance benchmark for our automatic recognition system. Our test consisted of 140 audio clips from 14 categories, with ten clips from each of the classes described in Section V-A. Audio clips were randomly picked from the test and training sets, and the duration varied between 2, 4, and 6 s. A total of 18 subjects participated in the test. They were volunteers and had no prior experience in analyzing environmental sounds. Participants consisted of both male and female subjects with their ages between 24–40. About half of the subjects were from academia while the rest were from nonrelated fields. Four of the subjects were involved in speech and audio research.

Each subject was asked to complete 140 classification tasks (the number of audio clips) in the course of this experiment. In each task, subjects were asked to evaluate the sound clip presented to them by assigning a label of one of 15 choices, which includes the 14 possible scenes and the *others* category. In addition to class labeling, we also obtained the confidence level for each of the tasks. The confidence levels were between 1 and 5, with 5 being the most confident. The order in which sound clips were presented was randomized to minimize any bias. The test was set up so that the first 14 clips were samples of each of the classes and was not included in calculating the final results. They were used to introduce subjects to the variety of sounds to be examined and to accustom them to different categories.

The user interface was a web page accessible via a browser with internet connection. Users were asked to use headphones so as to reduce the amount of possible background noise. The test was performed without any time limit, and users were able to break and return at any time. For each task, the results are expressed as an entry consisting of four data items: 1) the original environment type, 2) the audio clip duration, 3) user labeled environment type, and 4) user confidence level.

B. Test Results

The results from the listening test are shown in Fig. 11. The overall recognition rate was 82.3%, and the recognition accuracy for each individual environment ranged from 50% to 100%. The three best recognized scenes were *Nature-daytime* (98%), *Playground* (95%), and *Thundering* (95%). On the other hand, the four most difficult scenes were *Ocean waves* (65%), *Inside Casino* (70%), *Inside moving vehicles* (73%), and *Street with traffic* (74%). The listening test showed that humans are able to recognize everyday auditory scenes in 82% of the cases. The confusions were mainly between scenes that had similar types of prominent sound events. We can also examine the performance

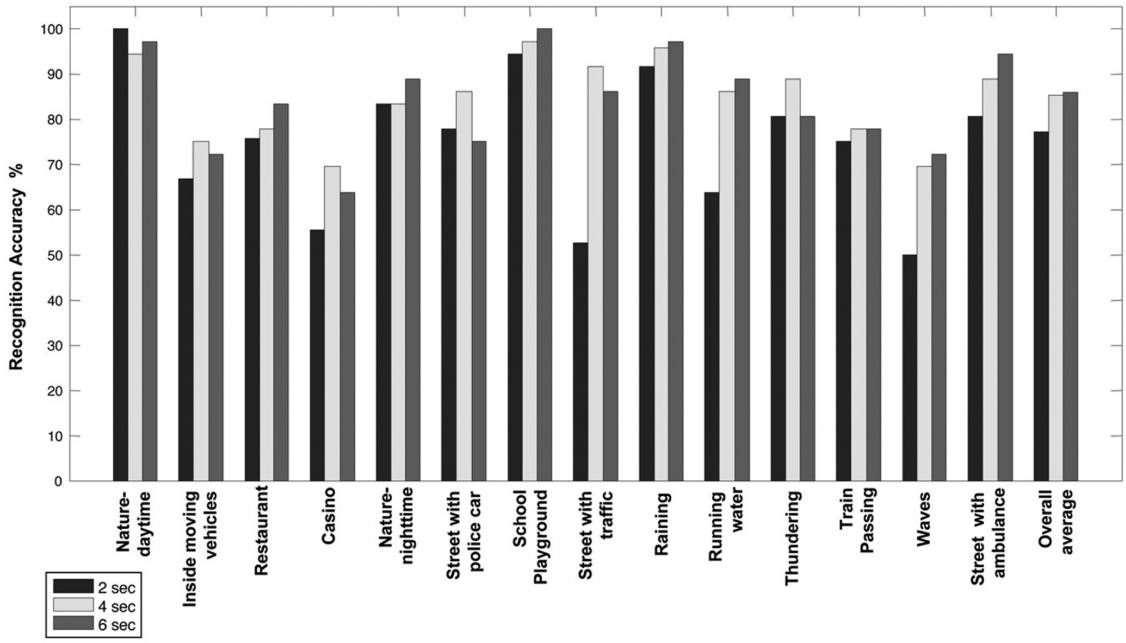

Fig. 11. Recognition accuracy of 14 classes from the listening test.

of each sound class as an effect of the duration in Fig. 11. The overall average recognition rates were 77%, 82%, and 85% for an audio clip duration of 2, 4 and 6 s, respectively. There is a larger difference in the rates between 2 and 4 s, but less between 4 and 6 s. A longer duration permits the listener more opportunities to pick up prominent sounds within each clip. However, the duration effect becomes less important as it passes a certain threshold.

One of the main reasons for misclassification was due to misleading sound events. For example, the scene *Street with traffic* was recorded with different types of traffic, which was frequently recognized as *Inside moving vehicles*, and vice versa. The recordings from *Inside moving vehicles* consist of different vehicles passing, which included a variety of vehicles like passenger sedans, trucks, and buses. Another reason for misclassification arises from the similarity between two different sounds and the inability of human ears to separate them. For example *Ocean waves* actually sounds very similar to that of *Train passing*. Another problem comes from subjects' unfamiliarity of a particular scene. For example, some users reported that they have never set foot inside a casino. Thus, the sound event *Inside casino* was mislabeled by them as *Inside restaurant* due to the crowd type of the ambient sound.

The confusion matrix for the test is given in Table VI. The rows of the matrix are the presented environmental scenes while the columns describe the subject responses. All values are given in percentages. Confusion between scenes was most noticeably high between *Street with police car* and *Streets with ambulance*, between *Raining* and *Running water*, and between *Street with traffic* and *Inside moving vehicles*.

The highest off-diagonal value occurs when *Streets with police car* is recognized as *Street with ambulance*. Confusion between sirens from police cars and ambulance was not due to the actual discrimination between the two sound classes but rather some people were semantically confused between the two sirens. In other words, the discrimination between the two classes requires background knowledge of subjects. Many users reported afterwards that they were second guessing the type of emergency vehicles that sirens were originating from. Confusion also occurred between scenes that are filled with crowded people, such as *Inside restaurant* and *Inside casino*.

Besides recognition accuracy, we are also interested in the relationship between the user confidence level and the audio clip duration. The results are shown in Fig. 12. If we compare Figs. 11 and 12, a lower confidence translates to a lower recognition rate, and vice versa. The confidence of listeners increases as we extend from 2 to 4 s, but there is only a slight increase from 4 to 6 s. The average confidence for each class, out of a possible 5, is 3.7, 4.2, and 4.4 for 2 s, 4 seconds, and 6 s, respectively. The lowest scores with the largest discrepancy between 2 and 4 s comes from the pair of *Waves* and *Street with traffic*. In general, a higher confidence is displayed with audio clips that are longer than 2 s.

The listening test shows that human listeners were able to correctly recognize 82% of ambient environment sounds for a duration of 4 s. Under the condition of 4-s clips, our automatic recognition system achieved a rate of 83%, which demonstrates that our recognition system has comparable performance to that of human listeners.

The results of our listening test and those in [11] are dissimilar. As indicated in the studies in [11], their results were higher for humans than that obtained from the computer system. Whereas in our case, the results were fairly similar between human and computer recognition. One possible reason for the differences is that their experimental setup was different than the one presented here, most notably in the length of the data presented to the subjects. The data presented to the users in our setup are the same segments as used in our automatic classifi-

TABLE VI
RECOGNITION PERFORMANCE FROM THE LISTENING TEST

	Nature-daytime	Inside vehicle	Restau-rant	Casino	Nature-nighttime	Street - police	Play-ground	Street - traffic	Thun-dering	Train	Rain / shower	Stream/ river	Waves	Street - ambu-lance	Others
Nature-Daytime	98.5				1.5										
Vehicle		73.1						10.0		1.5			3.1		12.3
Restau-rant			86.9	3.0				4.8							5.4
Casino			14.8	70.0			3.1								11.5
Nature-nighttime	6.2				88.5			1.5							4.0
Police		6.2				76.2								17.9	
Play-ground							95.0		3.1						2.0
Traffic			15.6					73.8			5.6				3.8
Thundering									94.6	2.3	3.1				
Train								8.5		80.8		3.1			4.8
Rain											82.3	15.6			2.3
Stream											16.9	83.9			
Waves								13.1	2.3		4.6	2.3	65.4		2.3
Ambu-lance						16.4								83.1	

All values are in percentages. Blank cells equates to less than 1%.

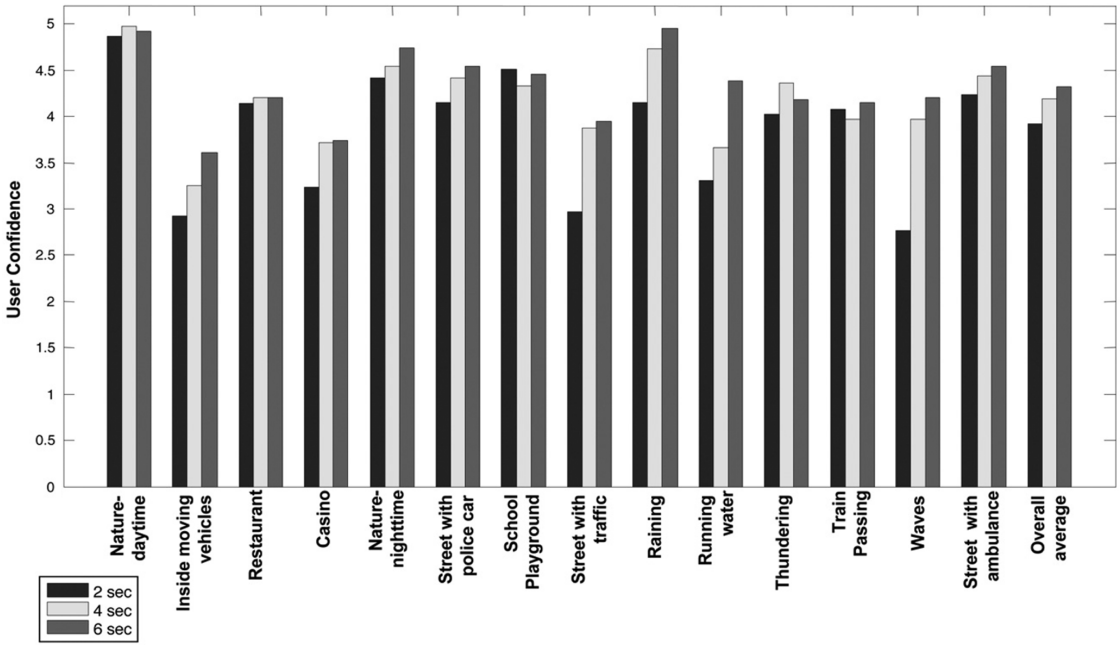

Fig. 12. User confidence in the listening test.

cation system, which was 4 seconds long, while the samples in Eronen's experiments were 30 s to 1 min long. Given that humans may have prior knowledge to different situations that can be advantageously used in classification, allowing them a much longer time to listen to the audio sample increases the likelihood that they would find some audio cue within each segment as to the environmental context in question.

VII. CONCLUSION

The paper reports a novel feature extraction method that utilizes matching pursuit (MP) to select a small set of time–frequency features, which is flexible, intuitive and physically interpretable. MP features can classify sounds where the pure frequency-domain features fail and can be advantageous combining with them to improve the overall performance. Extensive experiments were conducted to demonstrate the advantages of MP features as well as joint MFCC and MP features in environmental sound classification. The experimental results show promising performance in classifying 14 different audio environments, and shows comparable performance to human classification results on a similar task. Our work provides competitive performance for multi-audio category environment recognition using a comprehensive feature processing approach.

ACKNOWLEDGMENT

The authors would like to thank R. Pique-Regi and N. Cho for their helpful comments and suggestions.

REFERENCES

[1] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Special Iss. Socially Interactive Robots, Robot., Autonomous Syst.*, vol. 42, no. 3–4, pp. 271–281, 2003.

[2] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Minerva: A second generation mobile tour-guide robot," in *Proc. ICRA*, 1999.

[3] H. A. Yanco, "A robotic wheelchair system: Indoor navigation and user interface," in *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial Intelligence*. New York: Springer-Verlag, 1998, pp. 256–268.

[4] A. Fod, A. Howard, and M. J. Mataric, "Laser-based people tracking," in *Proc. ICRA*, 2002.

[5] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. J. Matarić, "Where am I? Scene recognition for mobile robots using audio features," in *Proc. ICME*, 2006.

[6] J. Huang, "Spatial auditory processing for a hearing robot," in *Proc. ICME*, 2002.

[7] A. Waibel, H. Steusloff, and R. Stiefelhagen, "Chil—Computers in the human interaction loop," in *Proc. WIAMIS*, 2004, and the CHIL Project Consortium.

[8] D. P. W. Ellis and K. Lee, "Minimal-impact audio-based personal archives," in *Proc. CARPE*, 2004.

[9] J. Mantyjarvi, P. Huuskonen, and J. Himberg, "Collaborative context determination to support mobile terminal applications," *IEEE Trans. Wireless Communications*, vol. 9, no. 5, pp. 39–45, Oct. 2002.

[10] D. P. W. Ellis, "Prediction-driven computational auditory scene analysis," Ph.D. dissertation, Dept. of Elect. Eng. and Comput. Sci., Mass. Inst. of Technol., Cambridge, MA, Jun. 1996.

[11] A. Eronen, V. Peltonen, J. Tuomi, A. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 321–329, Jan. 2006.

[12] R. G. Malkin and A. Waibel, "Classifying user environment for mobile applications using linear autoencoding of ambient audio," in *Proc. ICASSP*, 2005.

[13] V. Peltonen, "Computational auditory scene recognition," M.S. thesis, Tampere Univ. of Technol., Tampere, Finland, 2001.

[14] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *J. Acoust. Soc. Amer.*, vol. 122, no. 2, pp. 881–891, Aug. 2007.

[15] R. Cai, L. Lu, A. Hanjalic, H. Zhang, and L.-H. Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 1026–1039, May 2006.

[16] R. Radhakrishnan, A. Divakaran, and P. Smaragdis, "Audio analysis for surveillance applications," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, 2005, pp. 158–161.

[17] P. Cano, M. Koppenberger, S. Le Groux, J. Ricard, N. Wack, and P. Herrera, "Nearest-neighbor generic sound classification with a wordnet-based taxonomy," in *Proc. 116th AES Conv.*, 2004.

[18] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques," in *Proc. ICMAI*, 2002.

[19] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.

[20] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *Proc. ICASSP*, 1999.

[21] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *Proc. ICASSP*, 2000, pp. 149–152.

[22] T. Zhang and C.-C. J. Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 4, pp. 441–457, May 2001.

[23] R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 158–171, Feb. 1997.

[24] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, Jan. 2003.

[25] K. Umapathy, S. Krishnan, and S. Jimaa, "Multigroup classification of audio signals using time–frequency parameters," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 308–315, Apr. 2005.

[26] S. P. Ebenezer, A. Papandreou-Suppappola, and S. B. Suppappola, "Classification of acoustic emissions using modified matching pursuit," *EURASIP J. Appl. Signal Process.*, pp. 347–357, 2004.

[27] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition using mp-based features," in *Proc. ICASSP*, 2008, pp. 1–4.

[28] B. Clarkson, N. Sawhney, and A. Pentland, "Auditory context awareness via wearable computing," in *Proc. Workshop Perceptual User Interfaces*, 1998.

[29] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[30] G. Agostini, M. Longari, and E. Pollastri, "Musical instrument timbres classification with spectral features," *EURASIP J. Appl. Signal Process.*, vol. 2003, no. 1, pp. 5–14, 2003.

[31] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.

[32] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[33] S. Mallat and Z. Zhang, "Matching pursuits with time–frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[34] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals, Syst., Comput.*, 1993.

[35] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Muñoz, and F. López-Ferreras, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding," *IEEE Signal Process. Lett.*, vol. 11, no. 3, pp. 349–352, Mar. 2004.

[36] G. Yang, Q. Zhang, and P.-W. Que, "Matching-pursuit-based adaptive wavelet-packet atomic decomposition applied in ultrasonic inspection," *Russian J. Nondestructive Testing*, vol. 43, no. 1, pp. 62–68, Jan. 2007.

[37] P. Sugden and N. Canagarajah, "Underdetermined noisy blind separation using dual matching pursuits," in *Proc. ICASSP*, 2004, pp. 557–560.

[38] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of gaussian chirps," *IEEE Trans. Signal Processing*, vol. 49, no. 5, pp. 994–1001, May 2001.

[39] S. Ghofrani, D. McLernon, and A. Ayatollahi, "Comparing Gaussian and chirplet dictionaries for time–frequency analysis using matching pursuit decomposition," in *Proc. ISSPIT*, 2003.

[40] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[41] "The BBC Sound Effects Library—Original Series," [Online]. Available: http://www.sound-ideas.com/bbc.html

[42] "The Freesound Project," [Online]. Available: http://freesound.iua.upf.edu/index.php

[43] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 2003.

Selina Chu (S'06) received the B.S. degree from the Department of Electrical Engineering, California State Polytechnic University, Pomona, in 2000 and the M.S. degree from the Department of Computer Science from the University of California, Irvine, in 2002. She is currently pursuing the Ph.D. degree in the Department of Computer Science, University of Southern California (USC).

From 2002 to 2003, she was with the IBM T. J. Watson Research Center, Cambridge, MA. She was also a member of the Technical Staff at AT&T Labs-Research, Florham Park, NJ, for two summers in 1998 and 2000. Currently, she is a member of the Speech Analysis and Interpretation Lab (SAIL) and also the Multimedia Communications Lab. Her recent work has been in the areas of general unstructured audio. Her general research interests include audio signal processing, machine learning, data mining, and pattern recognition.

Ms Chu is a member of Tau Beta Pi and Eta Kappa Nu.

Shrikanth (Shri) Narayanan (M'95–SM'02–F'09) received the Ph.D. degree in electrical engineering from the University of California, Los Angeles (UCLA), in 1995.

He is Andrew J. Viterbi Professor of Engineering at the University of Southern California (USC), Los Angeles, and holds appointments as Professor of Electrical Engineering and jointly as Professor in computer science, linguistics, and psychology. Prior to USC, he was with AT&T Bell Labs and AT&T Research, first as a Senior Member, and later as a Principal Member, of its Technical Staff from 1995 to 2000. At USC, he is a member of the Signal and Image Processing Institute and directs the Signal Analysis and Interpretation Laboratory. He is an Editor for the *Computer Speech and Language Journal* (2007-present). He has published over 300 papers and has seven granted U.S. patents.

Prof. Narayanan is an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA. He was also an Associate Editor of the IEEE TRANSACTIONS OF SPEECH AND AUDIO PROCESSING (2000–2004) and the *IEEE Signal Processing Magazine* (2005–2008). He served on the Speech Processing technical committee (2005–2008) and Multimedia Signal Processing technical committees (2004–2008) of the IEEE Signal Processing Society and presently serves on the Speech Communication committee of the Acoustical Society of America and the Advisory Council of the International Speech Communication Association. He is a Fellow of the Acoustical Society of America and a member of Tau-Beta-Pi, Phi Kappa Phi, and Eta-Kappa-Nu. He is a recipient of an NSF CAREER award, a USC Engineering Junior Research Award, a USC Electrical Engineering Northrop Grumman Research Award, a Provost fellowship from the USC Center for Interdisciplinary research, a Mellon Award for Excellence in Mentoring, an IBM Faculty award, an Okawa Research award, and a 2005 Best Paper award from the IEEE Signal Processing society (with Alex Potamianos). Papers with his students have won best paper awards at ICSLP'02, ICASSP'05, MMSP'06, and MMSP'07.

C.-C. Jay Kuo (S'83–M'86–SM'92–F'99) received the B.S. degree from the National Taiwan University, Taipei, in 1980 and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively, all in electrical engineering.

He was a Computational and Applied Mathematics (CAM) Research Assistant Professor in the Department of Mathematics, University of California, Los Angeles, from October 1987 to December 1988. Since January 1989, he has been with the University of Southern California, where he is currently a Professor of Electrical Engineering, Computer Science, and Mathematics and a Director of the Signal and Image Processing Institute. His research interests are in the areas of digital signal and image processing, multimedia compression, communication, and networking technologies. Dr. Kuo has guided about 90 students to their Ph.D. degrees and supervised 20 postdoctoral research fellows. He is a coauthor of about 150 journal papers, 770 conference papers, and nine books. He is Editor-in-Chief for the *Journal of Visual Communication and Image Representation*, and Editor for the *Journal of Information Science and Engineering* and the *EURASIP Journal of Applied Signal Processing*.

Dr. Kuo is a Fellow of SPIE and a member of ACM. He was on the Editorial Board of the *IEEE Signal Processing Magazine* from 2003–2004. He served as Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING from 1995–1998, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 1995–1997, and IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING from 2001–2003. He received the National Science Foundation Young Investigator Award (NYI) and Presidential Faculty Fellow (PFF) Award in 1992 and 1993, respectively.