Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 607187, 12 pages
http://dx.doi.org/10.1155/2013/607187

Research Article

A Novel Element Detection Method in Audio Sensor Networks

Qi Li, Miao Zhang, and Guoai Xu

Key Laboratory of Network and Information Attack & Defense Technology of MOE, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Qi Li; qi.liqi2001@gmail.com

Received 14 June 2012; Accepted 19 December 2012

Academic Editor: Miguel-Angel Sicilia

Audio element detection in wireless sensor networks (WSNs) has great significance in our lives (e.g., in detecting traffic jam and accident, gun shots and explosion, and hurricane). It is particularly useful when video cameras cannot be used effectively (e.g., in darkness, with a wide range to cover); audio sensors are also much cheaper. However, most previous works on audio element detection require a large number of training examples to obtain satisfactory results. This becomes even more infeasible for audio sensors in WSNs where small energy consumption is required. In this paper, we propose a novel approach to solve this difficult problem. We first break down audio clips into a collection of simple "audio elements," and train these audio elements offline using statistical learning. Then, we train a weighted association graph using the trained audio element models online. This greatly reduces the amount of online training without sacrificing accuracy. We deploy our approach in an audio sensor network for traffic monitoring and venue monitoring to evaluate its performance. The experiments demonstrate that our proposed method achieves better results compared to the state-of-the-art methods while using smaller online training sets.

1. Introduction

Wireless sensor networks (WSNs) which combine the technology of sensor, embedded system, and wireless communications have a bright application prospect. At present, WSNs have been widely used in some specific area sensing and acquisition of scalar information, such as temperature and humidity. However, with the complexity and variation of monitoring demand, such scalar information cannot meet full demand of users. It is urgent to introduce rich media (i.e., audio, image, and video) into environment monitoring activities on the basis of sensor networks, thus performing a complete and accurate environment monitoring. Recently, multimedia sensor networks (MSNs) have been paid wide attention, and audio event detection in WSNs has great signification in our lives (e.g., in detecting traffic flow, jam, and accident) [1, 2].

Most smart home or smart traffic systems usually utilize video cameras to detect abnormal situations [3]. However, in some special situations, video cameras cannot work well. For example, the cameras are susceptible to poor weather conditions if outdoors, and they cannot work when black out. In addition to the video cameras, the use of audio sensors in surveillance and monitoring applications is becoming increasingly important [2, 4]. The audio sensors are particularly useful when video cameras cannot be used effectively (e.g., in darkness, with a wide range to cover). In some abnormal situations, audio conveys more significant information than video (e.g., human shouting/crying, car-crashing, etc.). Audio sensors are also much cheaper.

Audio element detection is important and helpful in many applications, such as audio scene analysis and audio summarization [5–7]. However, most relevant research focuses on detecting the audio elements in TV programs. In these works, most researchers collected enough training samples to train each audio element model. In practice, multiple audio elements may occur simultaneously in one audio clip. In certain controlled environment (such as movies), it is relatively easy to separate multiple audios by ICA (e.g., [3]). However, in real-world situations (such as on a noisy street), it is hard to separate multiple audios. Furthermore, building a model using training data is also difficult, due to a large number of training data required. Little previous work has solved this problem satisfactorily.

In this paper, we propose a novel approach to solve this difficult problem. We first break down audio clips into a collection of simple "audio elements." In this paper, audio elements are defined as short audio clips, and each represents a basic audio type (such as speech and music). We train these audio elements offline using statistical learning. Then, we train a weighted association graph using the trained audio elements online. This greatly reduces the amount of online training without sacrificing accuracy.

To further reduce the energy consumption of our detection method, we introduce a feature extraction method based on discriminating principle component analysis, DPCA. By using DPCA, we can reduce the dimension of the feature vector and distinguish different audio elements accurately.

The rest of this paper is organized as follows. In Section 2, we introduce some related works. In Section 3 we describe the system architecture briefly. Section 4 presents the feature extraction method. In Section 5, we present the weighted association graph-based element detection method. In Section 6, we show the experimental results. We conclude the paper and discuss the future works in Section 7.

2. Related Works

2.1. Feature Extraction. Feature extraction is one of the most fundamental and important issues in audio element detection. The general methodology of audio classification involves extracting discriminatory features from the audio data and feeding them to a pattern classifier [8].

Pfeiffer [9] presented a theoretical framework and application of automatic audio content analysis using some audio features in frequency domain. Nitanda et al. [10] presented a speech/music classifier based on simple features, such as zero-crossing rate and short-time energy for radio broadcast. Tzanetakis and Cook [11] proposed a novel method to classified music files into different musical genres by using low-level audio features. More specifically, the authors proposed three feature sets for representing timbral texture, rhythmic content, and pitch content. However, these simple features-based method focused more on music and speech signals and also cannot offer satisfactory results particularly when more audio classes are taken into consideration.

In recent years, a lot of audio features are introduced to detect more types of audio elements which include zero crossing rate [3], mel-frequency cepstral coefficients (MFCCs) [12], spectral similarity [8], and linear prediction coefficient-derived cepstral coefficients (LPCCs) [9]. However, we cannot use all the features to detect the audio events. On the one hand, the increase of the feature dimension may lead the increase of the computational complexity. On the other hand, the redundancy of the features may affect the detecting accuracy.

Some researchers reduce the feature dimension through the subspace analysis approaches. In pattern recognition, principal component analysis (PCA) [13] and linear discriminant analysis (LDA) [14] are the most popular subspace analysis approaches to learn the low-dimensional structure of high dimensional data, and they are widely used in the field of audio feature selection.

Principal component analysis, which is also known as Karhunen-Loeve (KL) transform, is a classical statistic technique that has been applied to many fields, such as knowledge representation, pattern recognition, and image compression. The objectives of PCA are to reduce the dimensionality of the dataset and identify new meaningful underlying variables. The key idea is to project the objects to an orthogonal subspace for their compact representations. However, the PCA does not pay any particular attention to the underlying class structure.

Linear discriminant analysis method searches for those vectors in the underlying space that best discriminate among classes (rather than those that best describe the data). More formally, given a number of independent features relative to which the data is described, LDA creates a linear combination of these which yields the largest mean differences between the classes. LDA is usually used in the fields of face recognition and auditory scene analysis for TV programs. However, in the audio sensor networks, the audio elements often happen in more complex situations and the samples of some special elements are difficultly acquired (e.g., car crashing in the streets). Moreover, the characteristic distributions of the testing data may have some differences with the training data.

Taking all of the data into account, PCA will compute a vector that has the largest variance associated with it. On the other hand, LDA will compute a vector which best discriminates between the classes. The PCA method is obviously of advantage to feature extraction but does not consider the separability of various classes. LDA can just make up for the deficiency of PCA. In this paper, we proposed a discriminating principle component analysis method which combines PCA and LDA together. The proposed method possesses the advantages of PCA and LDA while compensating the drawbacks of each individual; as a result, it can make good effect for feature extraction.

2.2. Audio Element Modeling and Detection. The audio element detection method can be divided into threshold-based method and statistical learning-based method.

Lu and Hankinson [15] used a threshold-based heuristic classification method to classify an audio signal into speech, music, and noise. For each feature, a threshold is set to determine the segment type and the feature set includes silence ratio, centroid, harmonicity, and pitch. However, since the feature threshold must change for different types of audio, the threshold-based classifier is tedious and not ideal.

Some statistical learning methods such as neural networks (NN) and support vector machines (SVMs) have been used to detect the audio elements. Mitra and Wang [7] extracted 26-D features including LPC and MFCCs to separate the audio clips into different audio types such as speech, music, and background noise. Artificial neural networks (ANNs), specifically multilayered perceptrons (MLPs), are implemented to perform the classification task. Lu et al. [16] proposed a content-based audio classification and segmentation method by using support vector machines. However, these classifiers are static classifiers, and they cannot deal with the continuous signal (e.g., audio signal) satisfactorily.

In [14, 15], several audio highlight events in the movies, such as applause, laughter, and cheer, are modeled by Hidden Markov Models (HMMs). In the detecting stage, audio features of each audio segment are extracted to be the inputs of these three highlight event models, and the highlight events in an audio clip are detected via a decision algorithm. However, these works were designed for TV programs and cannot be deployed on the audio sensor networks directly.

These methods had some common characteristics.

(1) In the above works, the audio stream is presegmented into some short audio clips with a given length. Then each audio clip will be recognized as a basic audio type. For example, in [16], the input audio stream is first segmented into units of 0.5 s with 0.125 s overlapping, then each unit is classified into applause, cheer, music, speech, and speech with music. However, in practice, multiple audio events may occur simultaneously, and a unit may contain several types of audio elements.

(2) The work using HMMs to detected audio elements usually classified the unit into the class which has the maximum log-likelihood score. However, in the monitoring systems, the target audio elements are usually sparsely distributed, and there are many nontarget sounds (e.g., the sound of speech in the traffic monitoring systems) which should be rejected in detection. Furthermore, the detection errors for different audio elements have different risk, for example, the missed detection risk for the sound of explosion is much higher than that of speech. We should find a method to minimize the detection risk as much as possible.

To solve the first problem, we propose a weighted association graph-based element detection method to detect the audio element in audio sensor networks. By using this method, one audio clip is considered as a series of audio elements, rather than as only one audio element.

To solve the second problem, we introduce the minimum risk Bayesian decision to solve the second problem. By this way, we can exclude the nontarget sounds from the target key element sequence and minimum the decision risk. Moreover, since it is easy for a single audio senor to make detection errors, we combine the sensory information of the same area by using a data fusion algorithm to increase the detection accuracy.

3. System Description

As shown in Figure 1, the audio stream can be divided into several audio clips according to the time duration. In practice, multiple audio elements may occur simultaneously in one audio clip. In certain controlled environment (such as movies), it is relatively easy to separate multiple audios by ICA (e.g., [3]). However, in real-world situations (such as on a noisy street), it is hard to separate multiple audios. Furthermore, building a model using training data is also difficult, due to a large number of training data required. Little previous work has solved this problem satisfactorily. Thus, we propose a weighted association graph-based element detection method to detect the audio element in the audio sensor networks.

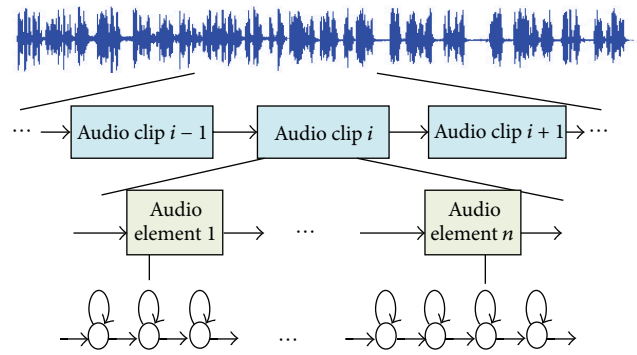


Figure 1: The segmenting method for an audio stream.

In this paper, audio elements are defined as short audio clips, and each represents a basic audio type (such as speech and music). In this method, we introduce the weighted association graphs to detect the audio element in audio sensor networks. First, we train the commonly occurred audio elements by HMM, respectively. Then, the transition probabilities between the basic audio elements were set by some specific rules. It actually constructs a higher-level probabilistic model, weighted association graph model, for audio event detection. The advantage of the weighted association graph is that we can effectively train each component separately. Moreover, this scheme keeps the advantage of framework flexibility in various applications. That is, when new basic audio effects are added in or removed from the association graphs, only the graphs need to be redefined, without any extra system retraining.

Feature extraction is one of the basic problems in audio element detection. The features should provide sufficient representations of the audio elements, as well as adequate discrimination among various basic audio elements. In the weighted association graph-based audio element detection method, a discriminating principal component analysis feature extraction method, DPCA, is proposed to improve the description of each basic audio element and to distinguish different audio elements.

As show in Figure 2, to detect the audio element, the audio sensors capture the audio signal in the environment, and then the feature vector of each frame, extracted by using the DPCA method, is passed through the above hierarchical structure. The log-likelihood scores of the audio clip with respect to the weighted association graphs and the optimal sequence of the basic audio elements are then calculated by the Viterbi algorithm. Then, the cluster head fuses the information collected by the sensor nodes in the cluster and make the final decision by using the minimum risk Bayesian decision algorithm.

4. Discriminating Principal Component Analysis Feature Extraction Method

Feature selection is one of the most fundamental and important issues in audio element detection. In the audio sensor

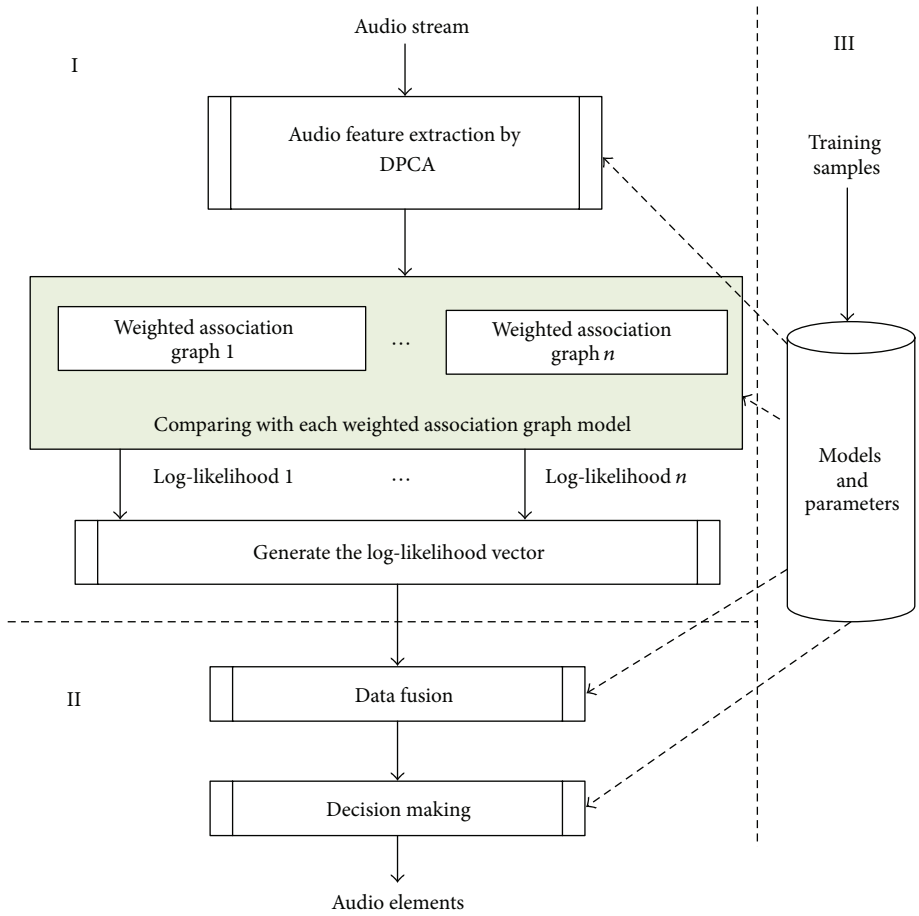


Figure 2: The process of the weighted association graph-based audio element detection method. It is mainly composed of three parts: (I) the working process of the audio sensor node; (II) the working process of the cluster head; (III) the off-line training process.

networks, we should limit the dimension of the feature vector and reduce the redundancy of the features to save the posterior computing energy. In this section, we introduce a discriminating principal component analysis feature extraction method to further reduce the energy consumption of our audio element detection method.

4.1. Basic Audio Features. Many audio features have been proposed in previous works on content-based audio analysis [1, 3, 4] and have been proved effective in detecting various key audio elements. Based on these works, in this paper, both temporal features and spectral features are extracted for each audio frame. The temporal features consist of short-time energy (STE) and zero-crossing rate (ZCR), and the spectral features consist of band energy ratios (BERs), frequency centroid, bandwidth, and Mel-frequency cepstral coefficients (MFCCs). In this paper, the spectral domain is equally divided into four frequency intervals, and the energy in each sub-band is then normalized by the whole spectrum energy. After Fourier transformation, frequency centroid and bandwidth are related to the first and second-order statistics of the spectrogram, respectively. MFCCs are sub-band energy features in Mel-scale, which give a more accurate simulation of the human auditory system. In this paper, 16-order MFCCs and its first-order differential coefficient (ΔMFCC) are extracted for the audio element detection.

4.2. Discriminating Principal Component Analysis. Given a dataset that consists of n audio samples, $X = \{x_1, x_2, \ldots, x_n\}$, the samples can be divided into C different audio classes.

The total scatter matrix of the projected samples, S_t, is defined as the following:

$$S_t = \frac{1}{n}\sum_{i=1}^{n} (x_i - u)(x_i - u)^T, \quad (1)$$

where u represents the mean feature vector of all samples in the training set and x_i is the ith sample's feature vector.

The between-class scatter matrix, S_b, is defined as the following:

$$S_b = \sum_{i=1}^{c} n_i (u_i - u)(u_i - u)^T, \quad (2)$$

where u_i is the mean feature vector of the samples in class i, and n_i is the number of samples in class i.

The within-class scatter matrix, S_w, is defined as the following:

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{n_j} \left(x_{ij} - u_i\right)\left(x_{ij} - u_i\right)^T, \tag{3}$$

where, x_{ij} is the jth sample in class i.

It follows from the definition that $S_t = S_b + S_w$.

(1) Objective function

The objective function of PCA is defined as follows:

$$W_{\text{opt}} = \arg\max_W \left(J_p(W)\right),$$
$$J_p(W) = \text{tr}\left[\left(W^T W\right)^{-1}\left(W^T S_t W\right)\right]. \tag{4}$$

The objective function of LDA is defined as follows:

$$W_{\text{opt}} = \arg\max_W \left(J_l(W)\right),$$
$$J_l(W) = \text{tr}\left[\left(W^T S_w W\right)^{-1}\left(W^T S_b W\right)\right]. \tag{5}$$

In this paper, we introduce the discriminating principal component analysis to extract the pivotal and independent audio features. The objective function of DPCA is defined as follows:

$$J_A(A) = \text{tr}\left[\left(A^T\left(S_w + \alpha I\right)A\right)^{-1} A^T S_b A\right], \tag{6}$$

$$A_{\text{opt}} = \arg\max_A \left(J_A(A)\right), \tag{7}$$

where α is an adaptive parameter.

When $\alpha = 0$, then

$$J_A(A) = \text{tr}\left[\left(A^T S_w A^{-1}\right)\left(A^T S_b A\right)\right], \tag{8}$$

and it is the objective function of LDA.

When α is large enough,

$$J_A(A) = \alpha \cdot \text{tr}\left[\left(A^T A\right)^{-1} A^T S_b A\right]. \tag{9}$$

α has no infection with the solution of A_{opt}, and $\text{tr}[(A^T A)^{-1} A^T S_b A]$ can be obtained by substituting $S_t = S_b$ into the objective function of PCA, and the recognition result by using S_b is similar to using S_t.

This method is the combination of PCA and LDA, which extracts the best representative audio features and enhance the determining ability through analyzing the categories of training samples. By using this method, we can extract the pivotal and independent features.

(2) Solution of A_{opt}

The crux of DPCA is finding the A_{opt} to maximize the value of $J_A(A)$.

That is,

$$\frac{\partial J_A(A)}{\partial A} = 0. \tag{10}$$

From (6) and (10), we have

$$-2\left(S_w + \alpha I\right)A\left[A^T\left(S_w + \alpha I\right)A\right]^{-1}\left(A^T S_b A\right) \times \left[A^T\left(S_w + \alpha I\right)A\right]^{-1} + 2S_b A\left[A^T\left(S_w + \alpha I\right)A\right]^{-1} = 0. \tag{11}$$

That is,

$$\left(S_w + \alpha I\right)^{-1} S_b A = A\left[A^T\left(S_w + \alpha I\right)A\right]^{-1}\left(A^T S_b A\right). \tag{12}$$

Then, we diagonalize $A^T(S_w + \alpha I)A$ and $A^T S_b A$ through the following linear transformations:

$$B^T A^T\left(S_w + \alpha I\right)AB = \text{I},$$
$$B^T A^T S_b AB = D. \tag{13}$$

We have

$$A^T\left(S_w + \alpha I\right)A = \left(B^T\right)^{-1} B^T = BB^T,$$
$$A^T S_b A = \left(B^T\right)^{-1} DB^{-1} = BDB^{-1} = BDB^T. \tag{14}$$

From (12) and (14), we have

$$\left(S_w + \alpha I\right)^{-1} S_b AB = A\left[A^T\left(S_w + \alpha I\right)A\right]^{-1} \times \left(A^T S_b A\right)B = ABB^T\left(B^T\right)^{-1} DB^{-1}B = ABD. \tag{15}$$

Let $U = AB$, and substitute equation $A = UB^{-1}$ into (15):

$$\left(S_w + \alpha I\right)^{-1} S_b U = UD. \tag{16}$$

It is a typical eigenvector-eigenvalue problem.

Since,

$$\begin{aligned} J_A(U) &= \text{tr}\left\{\left[U^T\left(S_w + \alpha I\right)U\right]^{-1} U^T S_b U\right\} \\ &= \text{tr}\left\{\left[B^T A^T\left(S_w + \alpha I\right)AB\right]^{-1} B^T A^T S_b AB\right\} \\ &= \text{tr}\left\{B^{-1}\left[A^T\left(S_w + \alpha I\right)A\right]^{-1}\left(B^T\right)^{-1} B^T A^T S_b AB\right\} \\ &= \text{tr}\left\{B^{-1}\left[A^T\left(S_w + \alpha I\right)A\right]^{-1} A^T S_b AB\right\} \\ &= \text{tr}\left\{\left[A^T\left(S_w + \alpha I\right)A\right]^{-1} A^T S_b ABB^{-1}\right\} \\ &= \text{tr}\left\{\left[A^T\left(S_w + \alpha I\right)A\right]^{-1} A^T S_b A\right\} = J_A(A). \end{aligned} \tag{17}$$

Let $T = (S_w + \alpha I)^{-1} S_b$, then we can obtain A_{opt} as follows:

$$A_{\text{opt}} = \left(v_1, v_2, \ldots, v_M\right), \tag{18}$$

where $Tv_i = \lambda_i v_i$ $(i = 1, 2, \ldots, N)$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$.

Then the original N-dimensional feature space is mapped to a M-dimensional feature space ($M < N$) through the linear transformation A_{opt}:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} v_1^T x \\ v_2^T x \\ \vdots \\ v_M^T x \end{bmatrix} = A_{\text{opt}}^T X. \quad (19)$$

(3) Determining the value of α

The value of the adaptive parameter α is a crucial issue in DPCA. According to extensive experiments, we obtain that the effects on experimental results caused by variation of α can be neglected when the value of α is in a proper range.

In this paper, we set $\alpha = [\text{tr}(S_b) + \text{tr}(S_w)]/M$.

5. Audio Event Modeling and Detection

Model establishing is extremely critical to the detection performance. In practice, multiple audio elements may occur simultaneously. In certain controlled environment (such as movies), it is relatively easy to separate multiple audios by ICA (e.g., [3]). However, in complex real-world situations (such as on a noisy street), it is hard to separate multiple audios. Furthermore, building a huge model using training data is also difficult, due to a large number of training data required. Little previous work has solved this problem satisfactorily. Thus, we propose a weighted association graph-based method to detect the element in audio sensor networks, in which we train the commonly occurred audio elements, respectively. Then, the transition probabilities between the basic audio elements were set by some specific rules. The advantage of this method is that we can effectively train each component separately. Moreover, this scheme keeps the advantage of framework flexibility in various applications. That is, when new basic audio effects are added in or removed from the event groups, only the graphs need to be redefined, without any extra system retraining.

5.1. Basic Audio Element Modeling. Since Hidden Markov Model has proved effective in many previous works for audio classification [4, 11], in this paper, we choose HMM for audio elements modeling and select the left-to-right structure as the topology of each HMM.

The HMM model for the ith basic audio element (BE_i) is characterized as follows:

$$H_i = \left(N_i, M_i, A_i, B_i, \Pi_i\right), \quad (20)$$

N_i is the number of states in the model of BE_i, M_i is the number of distinct observation symbols per state, A_i is the state transition probability distribution matrix, B_i is the probability distribution matrix of the observation symbol, and Π_i is the initial state distribution matrix.

Model size is a crucial issue in modeling HMMs. The state number should be large enough to characterize the variations of features. On the other hand, we need to simplify the computational complexity of model training and detection processes. We collect enough samples to estimate a reasonable model size of each audio element.

In this paper, we build the HMM for 12 basic audio elements, they are: car-engine, bus-engine, klaxon, car-braking, crashing, siren, speech, music, applause, laughter,s and cheer. The state number is set as shown in Table 1. These results make sense because we elaborately collect various kinds of samples for each audio element, and the experiments show the state number setting can achieve satisfied detection accuracy.

For each basic audio element, some short audio clips with the length of 3–10 seconds are selected as the training data. During training, the parameters for each state of an audio model are estimated by parsing the feature vectors of the training set. The Baum-Welch algorithm is then applied to estimate the transition probabilities between states and the observation probabilities in each state.

5.2. Weighted Association Graph. We notice that some basic audio elements usually occur together, while others seldom happen simultaneously. For instance, the sound of car-crashing often happens with car-braking, but rarely takes place with the sound of laugher. In some previous studies, the researchers have use the audio elements to indicate some special audio scene. For example, in [4, 5], car-engine, bus-engine, and klaxon are associated with traffic scene, and crashing and siren are used to indicate the abnormal events. Based on the above analysis, we introduce some audio-weighted association graphs to store the basic elements that always happen together.

According to [4, 5, 8], we divide the audio elements into two types: key audio element and background audio element. The key audio elements and the background audio element are set according to the monitor environment. We label the audio element which we focus on as the key audio element, and label the element we do not pay much attention to as the background elements.

A weighted association graph is composed of several key audio elements and background audio elements.

For a given graph G, let V be the set of the basic elements which always happen together:

$$V = \left\{BE_1, BE_2, \ldots, BE_N\right\}, \quad (21)$$

where BE_i is the ith basic audio element in the set. Then, we define the weighted association graph model as follows:

$$G = (V, E), \quad (22)$$

where $E = \{\langle BE_i, BE_j\rangle \mid BE_i, BE_j \in V \text{ and } p_{ij}\}$, where p_{ij} is the transition probability from BE_i to BE_j. Next, we will device the value p_{ij} in detail.

In this paper, we define 9 weighted association graphs based on 11 basic elements. The weighted association graphs and the related basic audio elements are listed in Table 2.

We assume that the following.

(1) A key audio element can only transfer to the basic audio elements in the same graph.

Table 1: The number of HMM states of the basic audio elements.

Traffic		Ceremony	
Basic audio element	NS	Basic audio element	NS
Car-engine	3	Speech	4
Bus-engine	3	Music	5
Klaxon	4	Laugher	4
Car-braking	3	Cheer	3
Crashing	4	Applause	3
Siren	3	Glass-broken	3

Table 2: The weighted association graphs and the related basic audio elements.

	Key elements	Background elements
T1	Klaxon	Car-engine, bus-engine
T2	Car-braking, crashing	Car-engine, bus-engine
T3	Siren	Car-engine, bus-engine
T4	—	Car-engine, bus-engine
C1	Applause	Speech, music
C2	Cheer	Speech, music
C3	Laugher	Speech, music
C4	Glass broken	Speech, music
C5	—	Speech, music

(2) A background element can transfer to the key audio elements in the same graph and other background elements.

(3) One basic audio element can belong to several weighted association graphs.

Given a basic audio element BE_i, we define its subsequent set, $\Phi(BE_i)$, as a set of all the basic audio elements which BE_i can transit to, that is,

$$\Phi\left(BE_i\right) = \bigcup_{k|BE_i \in G_k} G_k, \tag{23}$$

where G_k is the kth weighted association graph that BE_i belongs to.

To avoid the training problem and enhance the detection flexibility, in this paper, the transition probabilities from BE_i to the audio elements in its subsequent set are set to be the same.

Thus, for a given basic audio element BE_i, for all $BE_j \in \Phi(BE_i)$, the transition probability from BE_i to BE_j can be set as follows:

$$p_{ij} = \frac{1}{\left|\Phi\left(BE_i\right)\right|}. \tag{24}$$

To this end, we have built the model for each audio weighted association graph by connecting the audio effect models through some specific rules. Figure 3 gives an example of the model for the weighted association graph T2.

In the testing process, the audio frames are estimated by each weighted association graph model, the Viterbi algorithm is used to compute the most likely state sequence for each model and the log-likelihood value with respect to each weighted association graph will be calculated for each frame. The vector composed of the log-likelihood values is used as the input of the data fusion stage.

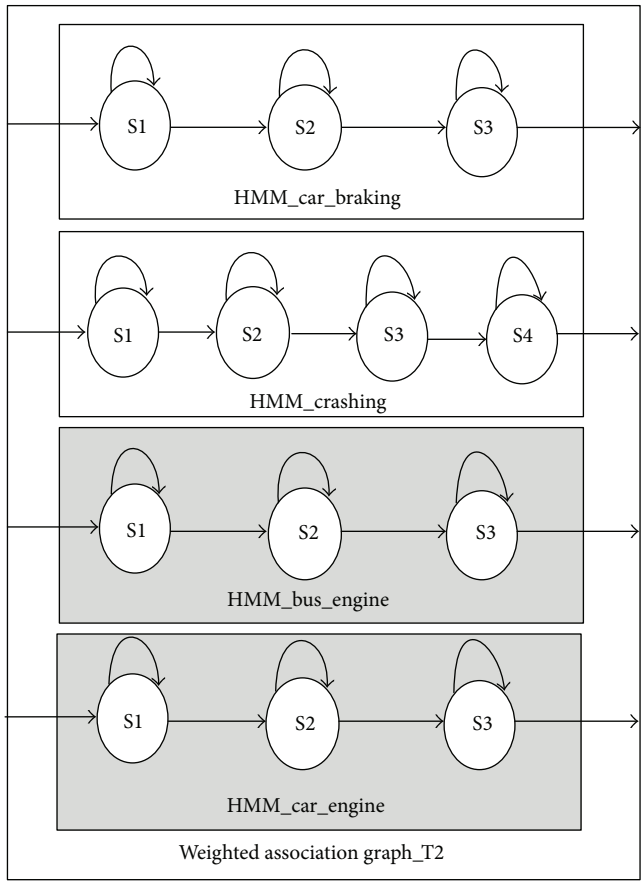


Figure 3: The model of the weighted association graph T2.

5.3. Data Fusion and Decision Making. After collecting information from the sensor nodes, the cluster head should fuse the information collected in the same monitoring area.

Consider a cluster with N ($N > 1$) audio sensors. Let s_i be the vector of n log-likelihood collected by the ith audio sensor node:

$$s_i = \left[s_{it}, s_{i2}, \ldots, s_{im}\right]^T, \tag{25}$$

where m is the number of the weighted association graphs we paid attention to, s_{ij} is the log-likelihood score under the model of the jth weighted association graph.

After collecting information from all the sensor nodes of the monitoring area, the cluster head should fuse the sensory information as follows:

$$\begin{aligned} s_{\text{fused}} &= \left[f_1, f_2 \ldots, f_m\right]^T, \\ f_i &= \sum_{k=1}^{N} \alpha_k \cdot s_{ik}, \end{aligned} \tag{26}$$

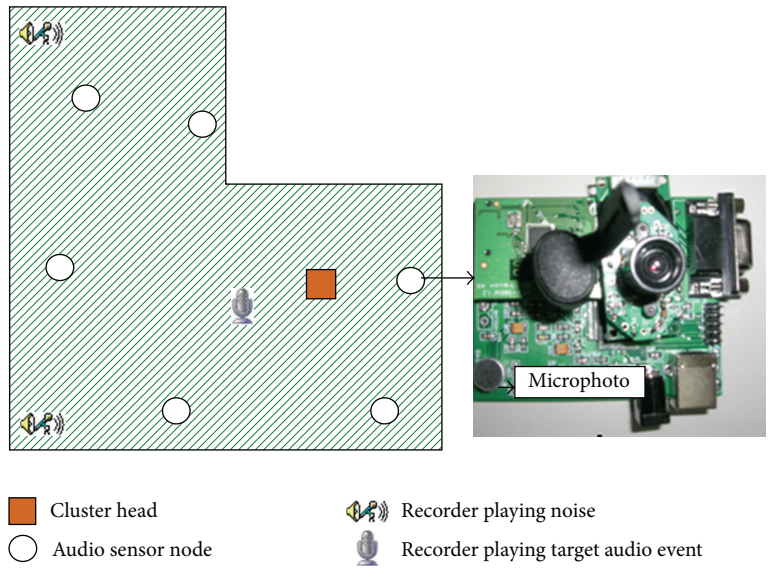


Figure 4: The structure of the selected cluster.

where α_k is the weight of the kth sensor node and the weights satisfy

$$\sum_{k=1}^{N} \alpha_k = 1. \tag{27}$$

In this paper, the weight values are measured by experiment data.

After data fusion, the fused log-likelihood value vector will be calculated for each audio frame. Then, the most important issue is how to make decision based on the log-likelihood vector. In the monitoring systems, the target audio elements are usually sparsely distributed, and there are many nontarget sounds (e.g., the sound of speech in the traffic monitoring systems) which should be rejected in detection. Furthermore, the missed detection for different audio elements have different risk. For example, the missed detection for the sound of explosion has the higher risk than that of speech. Based on the above analysis, we cannot simply classify the audio clip into the weighted association graph which has the maximum log-likelihood score. In this paper, an optimal decision is made based on the minimum risk Bayesian decision theory.

Let x be observed audio clip and s_{fused} be its log-likelihood value vector after data fusion.

We define the following.

w_{i1}: x belongs to G_i.

w_{i2}: x does not belong to G_i.

α_{i1}: determine x belong to G_i.

α_{i2}: determine x doesn't belong to G_i.

Let $\lambda(j,k)$ be the risk factor for making the decision of α_k when the fact is w_j.

Then, the risk for the decision α_{i1} can be described as the following:

$$R_{i1}(x) = \lambda_i(1,1) P(w_{i1} \mid f_i) + \lambda_i(2,1) P(w_{i2} \mid f_i). \tag{28}$$

The risk for the decision α_{i2} can be described as the following:

$$R_{i2}(x) = \lambda_i(1,2) P(w_{i1} \mid f_i) + \lambda_i(2,2) P(w_{i2} \mid f_i). \tag{29}$$

We define C_i as the decision factor of the audio clip x under the weighted association graph G_i,

$$\begin{aligned} C_i &= \frac{R_{i2}}{R_{i1}} \\ &= \frac{\lambda_i(1,2) P(w_{i1} \mid f_i) + \lambda_i(2,2) P(w_{i2} \mid f_i)}{\lambda_i(1,1) P(w_{i1} \mid f_i) + \lambda_i(2,1) P(w_{i2} \mid f_i)}. \end{aligned} \tag{30}$$

It is obvious that, if $C_i < 1$, the risk of α_{i1} is larger than the risk of α_{i2}, and the larger C_i is the more probability of α_{i1}. Then, we will introduce how to calculate C_i in detail.

Since $\lambda(i,i) = 0$, (30) can be rewritten as the following:

$$C_i = \frac{\lambda_i(1,2) P(w_{i1} \mid f_i)}{\lambda_i(2,1) P(w_{i2} \mid f_i)}. \tag{31}$$

$P(A \mid B)$ is the posteriori probability, which can be denoted by priori probability $P(A)$ and the conditional probability $P(A \mid B)$:

$$P(A \mid B) = P(B \mid A) P(A). \tag{32}$$

So (31) can be rewrite as the followings:

$$C_i = \frac{\lambda_i(1,2) P(f_i \mid w_{i1}) P(w_{i1})}{\lambda_i(2,1) P(f_i \mid w_{i2}) P(w_{i2})}. \tag{33}$$

Table 3: Equipment specification.

Parameter	Value
CPU	72 MHZ
FLASH	256 KB
SRAM	64 KB
Sampling rate	8 KHZ

$P(w_1)$, $P(w_2)$, $P(x \mid w_1)$, and $P(x \mid w_2)$ can be estimated by the priori information in the training database. Generally, both the distribution probability density functions of $P(x \mid w_1)$ and $P(x \mid w_2)$ are negative gamma distribution.

In this paper, after getting the log-likelihood value vector, we calculated the decision vector C:

$$C = [C_1, C_2, \ldots, C_m], \tag{34}$$

where m is the number of the weighted association graphs we paid attention to, C_i is the decision vector under the model of the ith weighted association graph, then the final decision as follows.

(1) The audio clip x is considered as non-predefined sound if $\max_i(C_i) < 1$.

(2) Otherwise, the audio clip x is classified into the ith weighted association graph with the largest decision factor:

$$i = \arg\max_j \left(C_j\right). \tag{35}$$

6. Implement and Evaluation

We set up an audio sensor network, consisting of 8 clusters including 48 sensor nodes in all, for the audio element detection. In this paper, we select one cluster of the network to analyze the detailed implementations and evaluations of the proposed method. The architecture of the selected cluster is shown in Figure 4.

As shown in Figure 5, we use a PC as the cluster head and the sensor nodes can communicate with the cluster head based on the ZigBee wireless communication protocol. The detail parameters of the sensor node are described in Table 3.

6.1. Evaluation of the Feature Extraction Method. In this experiment, we compare the DPCA feature extraction method with the PCA method and LDA method. To fully evaluate the performance of the three-feature extraction method, 12 basic audio elements are taken into account in the experiments. They are car-engine, bus-engine, klaxon, car-braking, crashing, siren, speech, music, applause, cheer, laugher, and glass broken. The selected audio elements frequently happen in the traffic scene or the celebration scene, and play important roles in humans' understanding of the high-level audio semantics.

We first extract the 41-D basic feature vector for each audio frame according to Section 4.1. Then we reduce vector dimension of the basic feature by using three different methods (PCA, LDA, and DPCA). Finally, the low-dimension

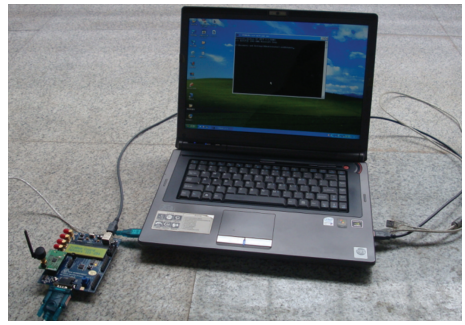


Figure 5: The cluster head of the audio sensor network.

feature vectors are sent to both the HMM-based recognizer and the SVM-based recognizer.

For each basic audio element, its precision and recall are defined as follows:

$$\text{precision} = \frac{n_c}{n_r}, \qquad \text{recall} = \frac{n_c}{n_t}, \tag{36}$$

where n_c denotes the number of correctly detected frames, n_r represents the number of all the frames recognized as the target element, and n_t is the total frame number of the target element in truth.

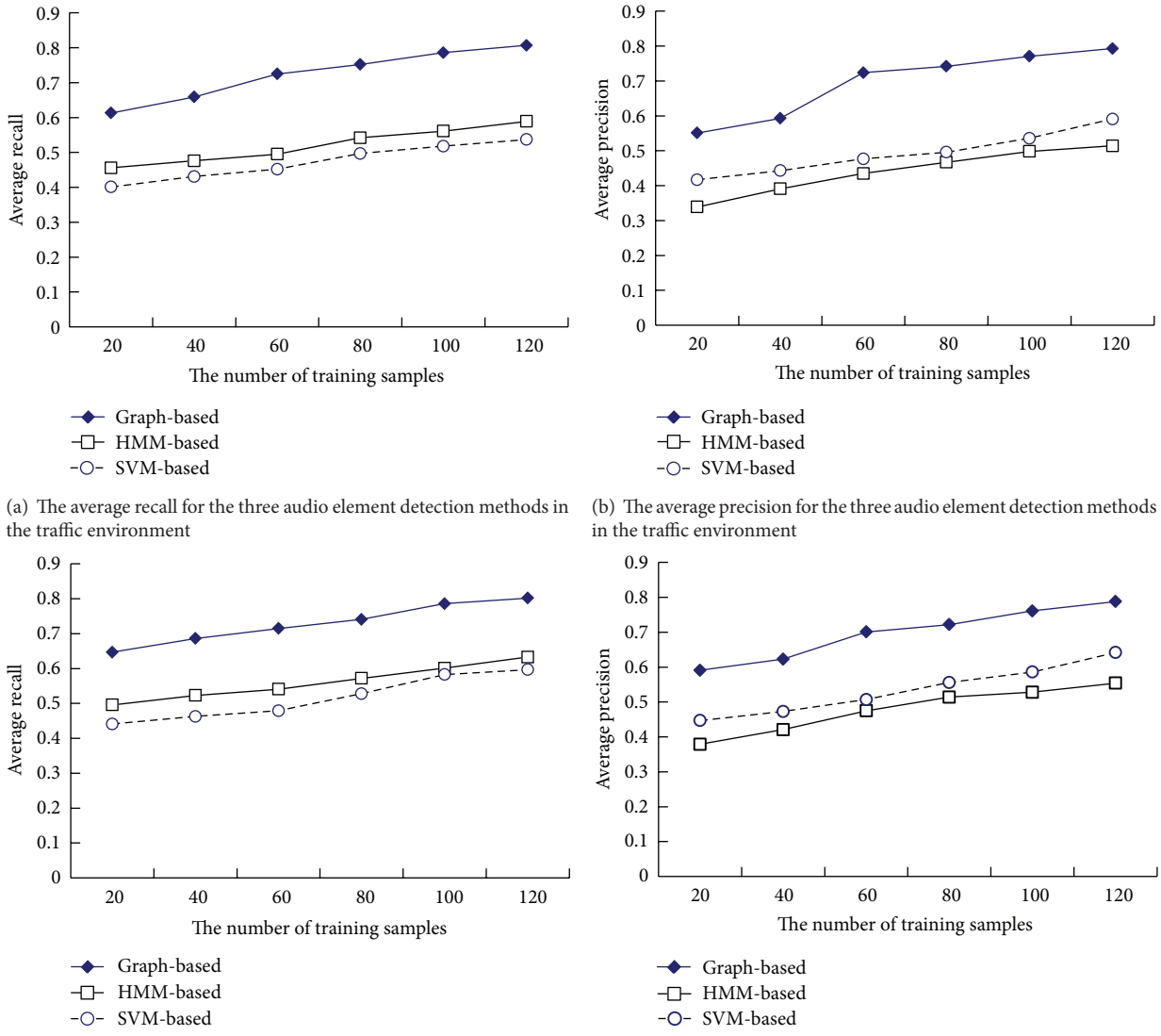
We change the target feature dimension and analyze the change of the average recall and average precision for the three feature extraction methods (Tables 4 and 5).

PCA is a classical multivariate analysis that is usually used to reduce the dimensionality of a dataset while retaining the beneficial information possibly. However, it does not pay any particular attention to the underlying class structure. When there are various types of audio elements and the target dimension of the feature vector is low, the PCA method cannot work well.

In theory, given large and representative learning data sets, LDA should outperform PCA. However, in the audio sensor networks, some specific audio elements often happen in more complex situations and the samples of the specific audio elements are difficultly acquired (e.g., car-crashing in the streets). Moreover, the characteristic distributions of the testing data may have some differences with the training data. In that situation, when the target dimension of the feature is low, LDA method cannot achieve satisfying detection accuracy.

The PCA method is obviously of advantage to feature extraction, but it does not considered for the separability of various classes. Aiming at optimal separability of feature space, LDA can just make up for the deficiency of PCA. DPCA combines PCA and LDA together and adjusts the weights of PCA and LDA through an adaptive parameter. The experimental results show that, the DPCA method possesses the advantages of PCA and LDA while compensating the drawbacks of each individual. By using this method, we can extract the pivotal and independent features.

6.2. Evaluation of the Weighted Association Graph Model. To fully evaluate the performance of the weighted association



(a) The average recall for the three audio element detection methods in the traffic environment

(b) The average precision for the three audio element detection methods in the traffic environment

(c) The average recall for the three audio element detection methods in the celebration environment

(d) The average precision for the three audio element detection methods in the celebration environment

Figure 6: The detection results for the three audio element detection methods.

graph model in audio element detection, 9 different kinds of weighted association graphs are selected for analysis. The weighted association graphs and the related basic audio element are listed in Table 2. The traffic data set (95.4 minutes) is collected on a busy road outside. The celebration data set (73.6 minutes) was collected in university hall. The experimental results are shown in Figure 6.

From Figure 6, we can see that because the real-world situations are more complex, and multiple audio events always happen together, so both the HMM-based method and SVM-based method need a large number of training data to build the audio event models. By using the weighted association graph based algorithm, we combine the statistical learning and human knowledge together and reduce the training complexity for complex real-world situations. When the number is the same, the weighted association graph-based algorithm can achieve higher detection accuracy than the other two methods. Moreover, this scheme keeps the advantage of framework flexibility in various applications. That is, when new basic audio elements are added in or removed from the association graphs, only the graphs need to be redefined, without any extra system retraining.

6.3. Evaluation of the Bayesian Decision Algorithm. In this experiment, we compare the Bayesian decision based-algorithm with the max-log-likelihood-based algorithm.

As shown in Table 6, for traffic environment monitor, the detection precision of using Bayesian decision is much better than that of max log-likelihood. For celebration, the detection precision of various audio elements is better than of max log-likelihood. Because there exists some non-predefined audio elements in real-world audio monitor, Bayesian decision can remove these non-predefined audio elements, which

Table 4: The average recall by using three different feature extraction methods (PCA, LDA, and DPCA).

	HMM			SVM		
	PCA	LDA	DPCA	PCA	LDA	DPCA
10	0.624	0.613	0.754	0.621	0.584	0.735
15	0.702	0.712	0.811	0.673	0.671	0.763
20	0.754	0.776	0.841	0.724	0.741	0.803
25	0.817	0.825	0.864	0.744	0.772	0.834
30	0.853	0.871	0.874	0.763	0.794	0.837

Table 5: The average precision by using three different feature extraction methods (PCA, LDA, and DPCA).

	HMM			SVM		
	PCA	LDA	DPCA	PCA	LDA	DPCA
10	0.671	0.637	0.714	0.673	0.662	0.723
15	0.683	0.684	0.738	0.724	0.721	0.744
20	0.742	0.747	0.804	0.767	0.776	0.812
25	0.775	0.792	0.809	0.792	0.813	0.844
30	0.803	0.816	0.816	0.834	0.839	0.848

Table 6: The audio element detection results by using Bayesian decision based algorithm and the max-log-likelihood-based algorithm.

Audio element	Max-log likelihood		Bayesian	
	Recall	Precision	Recall	Precision
Car-engine	0.713	0.722	0.751	0.776
Bus-engine	0.743	0.758	0.809	0.811
klaxon	0.691	0.754	0.742	0.803
Car-braking	0.662	0.673	0.771	0.781
Crashing	0.673	0.704	0.731	0.749
Siren	0.736	0.742	0.799	0.804
Glass-broken	0.680	0.704	0.744	0.753
Applause	0.753	0.764	0.792	0.787
Laugher	0.712	0.733	0.737	0.744
Cheer	0.691	0.731	0.735	0.768
Music	0.724	0.737	0.755	0.761
Speech	0.693	0.746	0.690	0.771

improves the precision of predefined audio elements and especially efficient ones in the traffic monitor. The key elements usually occur between the background elements, and the last time is short. As a result, the recall of key elements by using Bayesian is much better than that of max log-likelihood. For example, the last time of crash is 0.3–0.5 seconds, which is much shorter than the length of detecting windows. If we use the max log-likelihood to detect the crash, the sampling window is considered as background element, while the Bayesian decision can solve this problem.

7. Conclusions and Future Works

In this paper, we propose a weighted association graph-based element detection method to detect the audio elements in the audio sensor networks. In the proposed method, we train the basic audio elements separately based on statistical learning and combine them together by some prior knowledge in specific domains. By this method we can combine the statistical learning and human knowledge together and reduce the training complexity for the real-world situations. Moreover, a discriminating principle component analysis based feature extraction method is introduced to improve the representation of each audio element and the discrimination among various elements. Finally, we introduce the minimum risk Bayesian decision to reduce the decision risk. We deploy this method on an audio sensor network to evaluate its performance, and the experiment evaluations demonstrate that the proposed method can achieve satisfied results.

At present, sensed data are gradually transformed to semantic web data. Therefore, how to organize and manage the sensed data becomes an urgent matter. We think that audio semantic extraction may fill this gap to accelerate transformation from sensed audio data to the Semantic Web data. Moreover, the Semantic Web offers us a new approach to manage the sensed information. In our future work, we will mainly focus on two aspects: one is audio semantic analysis which is about how to support Semantic Web development in ontology learning and ontology query. The other is Semantic Web which is about how to improve the results in audio semantic extraction.

Acknowledgment

This work is supported by the National Science and Technology Projects (2012ZX03002012).

References

[1] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 1306–1309, Amsterdam, The Netherlands, July 2005.

[2] S. Moncrieff, S. Venkatesh, and G. West, "Online audio background determination for complex audio environments," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 3, no. 2, Article ID 1230814, 2007.

[3] N. Naikal, A. Y. Yang, and S. S. Sastry, "Towards an efficient distributed object recognition system in wireless smart camera networks," in *Proceedings of the 13th Conference on Information Fusion (Fusion '10)*, pp. 1–8, July 2010.

[4] M. Cristani, M. Bicego, and V. Murino, "On-line adaptive background modeling for audio surveillance," in *Proceedings of the IEEE 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 399–402, August 2004.

[5] Q. Li and H. Ma, "GBED: group based event detection method for audio sensor networks," in *Proceedings of the 17th ACM International Conference on Multimedia (MM '09) with Co-located Workshops and Symposiums*, pp. 857–860, Beijing, China, October 2009.

[6] R. Cai, L. Lu, A. Hanjalic, H. J. Zhang, and L. H. Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 3, pp. 1026–1038, 2006.

[7] V. Mitra and C. J. Wang, "Content based audio classification: a neural network approach," *Soft Computing*, vol. 12, no. 7, pp. 639–646, 2008.

[8] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1236–1246, 2007.

[9] S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," in *Proceedings of the 4th ACM International Multimedia Conference*, pp. 21–30, November 1996.

[10] N. Nitanda, M. Haseyama, and H. Kitajima, "Accurate audio-segment classification using feature extraction matrix," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, pp. III261–III264, October 2005.

[11] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

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

[13] X. D. Xie and K. M. Lam, "Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2481–2492, 2006.

[14] G. Dai and Y. Qian, "A gabor direct fractional-step LDA algorithm for face recognition," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, pp. 61–64, Taipei, Taiwan, June 2004.

[15] G. Lu and T. Hankinson, "'Technique towards automatic audio classification and retrieval," in *Proceedings of the 4th International Conference on Signal Processing Proceedings (ICSP '98)*, pp. 1142–1145, October 1998.

[16] L. Lu, H. Zhang, and S. Li, "Content-based audio classification and segmentation by using support vector machines," *ACM Multimedia Systems Journal*, vol. 8, no. 6, pp. 482–492, 2003.