Audio Classification Based on MPEG-7 Spectral Basis Representations

Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora, *Senior Member, IEEE*

***Abstract*—In this paper, we present an MPEG-7-based audio classification and retrieval technique targeted for analysis of film material. The technique consists of low-level descriptors and high-level description schemes. For low-level descriptors, low-dimensional features such as audio spectrum projection based on audio spectrum basis descriptors is produced in order to find a balanced tradeoff between reducing dimensionality and retaining maximum information content. High-level description schemes are used to describe the modeling of reduced-dimension features, the procedure of audio classification, and retrieval. A classifier based on continuous hidden Markov models is applied. The sound model state path, which is selected according to the maximum-likelihood model, is stored in an MPEG-7 sound database and used as an index for query applications. Various experiments are presented where the speaker- and sound-recognition rates are compared for different feature extraction methods. Using independent component analysis, we achieved better results than normalized audio spectrum envelope and principal component analysis in a speaker recognition system. In audio classification experiments, audio sounds are classified into selected sound classes in real time with an accuracy of 96%.**

***Index Terms*—Audio spectrum basis (ASB), audio spectrum projection (ASP), hidden Markov models (HMMs), independent component analysis (ICA), MPEG-7.**

I. Introduction

AUDIO SIGNALS contain a great deal of information that can be used for effective video indexing either alone or together with visual information, including environmental sounds, background noises, foley, animal sounds, speech sounds, and nonspeech utterances. For these reasons, audio classification and retrieval is an important and challenging research topic. An important step of audio classification is feature extraction. An efficient representation should be able to capture sound properties that are the most significant for the task, robust under various environments and general enough to describe various sound classes. Because the environmental sounds consist of multiple noisy and textured components as well as higher order structural components such as iterations and scatterings, they are generally much harder to characterize than speech and music sounds. The mel-frequency cepstral coefficients (MFCC) approach [1], which is widely used in automatic speech recognition, has been proposed to extract audio features. The MFCCs are perceptually motivated features based on the short-term Fourier transform (STFT). The power spectrum bins are grouped and smoothed according to the perceptually motivated mel-frequency scaling. Then the spectrum is segmented into a number of critical bands by means of a filter bank that typically consists of overlapping triangular filters. Finally, a discrete cosine transform (DCT) applied to the logarithm of the filter bank outputs results in vectors of decorrelated MFCC features. Foote [2] proposes the use of MFCC coefficients plus energy to construct a learning tree vector quantizer. In the Muscle Fish system by Wold *et al.* [3], statistical values including means, variances, and autocorrelations of several time- and frequency-domain measurements are used to represent various perceptual features such as loudness, brightness, bandwidth, pitch, and harmonicity. Guo *et al.* [4] computes a combination of two types of features: 1) perceptual features, composed of total power, subband powers, brightness, bandwidth, and pitch and 2) MFCC vectors. These features are applied to the AdaBoost [5] learning machine which is compared with the support vector machine (SVM) technique [6]–[8]. Recently, in [9] and [10], Casey described a generalized sound recognition framework in which decorrelated, dimension-reduced log-spectral features are used to train hidden Markov models (HMMs) for various sounds such as speech, explosions, laughter, and different instruments. His important idea is to use basis functions consisting of decorrelated features that contain the important information of a spectrum in order to project it into a low-dimensional representation. To attain a good performance in this framework, a balanced tradeoff between reducing the dimensionality of data and retaining maximum information content must be performed, as too many dimensions cause problems with classification while dimensionality reduction invariably introduces information loss. The MPEG-7 sound recognition tools [10]–[12] according to his proposal use decorrelated spectral features based on independent component analysis (ICA) [13] basis functions with HMMs [14] in order to apply uniformly to diverse source classification tasks with accurate performance. The tools provide a unified interface for automatic indexing of audio using trained sound class models in a pattern recognition framework. Each classified audio piece will be individually processed and indexed so as to be suitable for efficient comparison and retrieval by the sound recognition system.

In this paper, our purpose is to evaluate the efficiency of an audio indexing and retrieval system based on audio spectrum basis (ASB) and audio spectrum projection (ASP) of MPEG-7 audio descriptors.

The paper is structured as follows. The audio indexing and retrieval system is described in Section II. Section III deals with evaluation of the automatically classified audio signals with or

Manuscript received April 30, 2003; revised September 4, 2003. This work was supported by the German Federal Ministry of Education and Research.

The authors are with the Communication Systems Group, Technical University of Berlin, 10587 Berlin, Germany (e-mail: kim@nue.tu-berlin.de; moreau@nue.tu-berlin.de; sikora@nue.tu-berlin.de).

Digital Object Identifier 10.1109/TCSVT.2004.826766

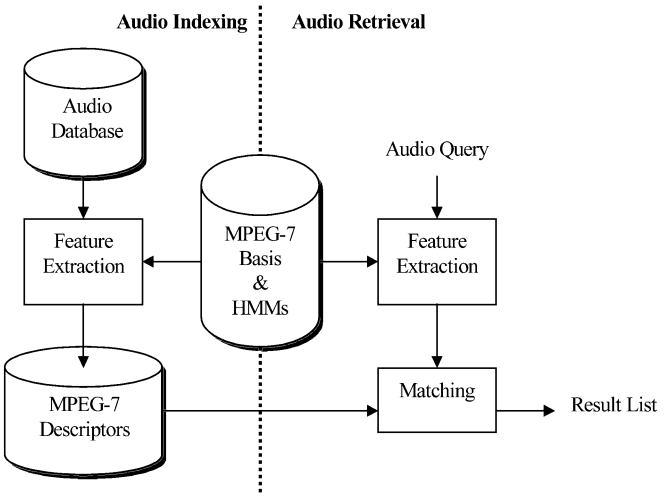

Fig. 1. Structure of audio indexing and retrieval system.

without a hierarchical structure consisting of various classes. Section IV presents conclusions and future directions.

II. Audio Indexing and Retrieval System

The structure of the audio indexing and retrieval system using MPEG-7 basis projection descriptors is illustrated in Fig. 1.

The audio indexing module extracts audio information from a database of sounds. An HMM and a basis function have been previously trained for each predefined sound class. A classification algorithm finds the most likely class for a given input sound by presenting it to each of the HMMs (after projection on the corresponding basis functions) and by using the Viterbi algorithm. The HMM with the highest maximum-likelihood score is selected as the representative class for the sound. The algorithm also generates the optimal HMM state path for each model given the input sound. The state path corresponding to the most likely class is stored as an MPEG-7 descriptor in the sound indexing database. It will be used as an index for further query applications.

The audio retrieval is based on the results of the audio indexing. For a given query sound, the extracted audio features are used to run the sound classifier as described above. The resulting state path corresponding to the most likely sound class is then used in a matching module to determine the list of the most similar sounds whose own state path descriptions are stored in a precomputed sound indexing database.

A. Feature Extraction Using Basis Projection

The purpose of MPEG-7 feature extraction is to obtain from the audio source a low-complex description of its content. A balanced tradeoff between reducing the dimensionality of data and retaining maximum information content must be achieved. For these reasons, the MPEG-7 audio group has proposed a feature extraction method based on the projection of a spectrum into a low-dimensional representation using decorrelated basis functions.

The feature extraction system using basis projection is described in Fig. 2. It mainly consists of five functions: short-time Fourier transform (STFT), audio spectrum envelope (ASE), normalized audio spectrum envelope (NASE), basis decomposition algorithm—such as singular value decomposition (SVD) or ICA—and basis projection, obtained by multiplying the NASE with a set of extracted basis functions.

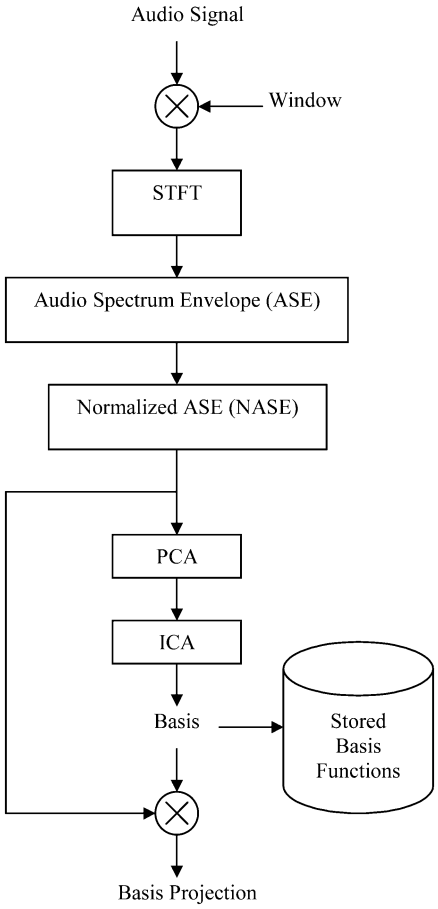

Fig. 2. Block diagram of feature extraction using spectrum basis projection.

For the basis decomposition step, we combined a basis dimension-reduction using a principal component analysis (PCA) algorithm [15] with a basis information maximization by ICA.

First, the observed audio signal $s(n)$ is divided into overlapping frames by the application of a hamming window function and analyzed using the STFT

$$S(l,k) = \sum_{n=0}^{N-1} s(n+lM)\, w(n) \exp\left(-j\left(\frac{2\pi}{N}\right)nk\right) \quad (1)$$

where N is the size of the STFT, $k\,(0 \le k \le K-1)$ is the frequency bin index, l is the time frame index, w is an analysis window of size lw, and M is the hop size. By Parseval's theorem (i.e., so that power is preserved), there is a further factor of $1/N$ to equate the sum of the squared magnitudes of the STFT coefficients as

$$P(l,k) = \frac{1}{nf \cdot N} |S(l,k)|^2 \quad (2)$$

where the window normalization factor nf is defined as

$$nf = \sum_{n=0}^{lw-1} w^2(n). \quad (3)$$

To extract reduced-rank spectral features, the spectral coefficients $P(l,k)$ are grouped in logarithmic subbands. Frequency channels are logarithmically spaced in nonoverlapping 1/4-octave bands spanning between 62.5 Hz ("low edge"), and 8 kHz ("high edge"). The output of the logarithmic frequency range is the sum of the power spectrum in each logarithmic subband. The spectrum according to a logarithmic frequency scale, which the MPEG-7 standard refers to as ASE, consists of a coefficient representing power between 0 Hz and "low edge," a series of coefficients representing power in logarithmically spaced bands between "low edge" and "high edge," and a coefficient representing power above "high edge."

The resulting log-frequency power spectrum is converted to the decibel scale

$$D(l,f) = 10\log_{10}(\mathrm{ASE}(l,f)) \tag{4}$$

where f is the logarithmic frequency range.

Each decibel-scale spectral vector is normalized with the rms energy envelope, thus yielding a normalized log-power version of the ASE called NASE. The full-rank features for each frame l consist of both the rms-norm gain value R_l and the NASE vector $X(l,f)$ as follows:

$$R_l = \sqrt{\sum_{f=1}^{F}\left(10\log_{10}\{\mathrm{ASE}(l,f)\}\right)^2},\quad 1 \le f \le F \tag{5}$$

$$X(l,f) = \frac{10\log_{10}\{\mathrm{ASE}(l,f)\}}{R_l},\quad 1 \le l \le L \tag{6}$$

where F is the number of ASE spectral coefficients and L is the total number of frames.

Much of the information is disregarded due to the lower frequency resolution when reducing the spectrum dimensionality from N to the F frequency bins of NASE.

In order to achieve a tradeoff between further dimensionality reduction and information loss, the ASB and audio spectrum projection (ASP) MPEG-7 low-level audio descriptors are used. To obtain the ASB, PCA or SVD [16], [17] and the more recently developed ICA perform high-dimension multivariate statistical analysis. PCA decorrelates the second-order moments corresponding to low-frequency properties and extracts orthogonal principal components of variations. ICA, on the other hand, is a linear but not necessarily orthogonal transform, which makes unknown linear mixtures of multidimensional random variables as statistically independent as possible. It not only decorrelates the second-order statistics but also reduces higher order statistical dependencies. It extracts independent components even if their magnitudes are small, whereas PCA extracts only components with the largest magnitudes. Thus, in the feature extraction process, the ICA representation captures the essential basis functions of the data.

Therefore, the next step of feature extraction in this paper is to extract a subspace from the NASE using a PCA algorithm. Then, to yield a statistically independent or uncorrelated component basis, the FastICA [18] algorithm is used. Some preprocessing such as centering and whitening is useful before using FastICA to estimate the uncorrelated basis functions matrix W. In the following, X will represent the input signal in the form of a $L \times F$ time-frequency matrix. The vertical dimension represents time (i.e., each row corresponds to a time frame index l), and the horizontal dimension represents the spectral coefficients (i.e., each column corresponds to a frequency range index f).

First, the columns should be centered by subtracting the mean value from each one as follows:

$$\hat{X}(f,l) = X(f,l) - \mu_f \tag{7}$$

$$\mu_f = \frac{1}{L}\sum_{l=1}^{L} X(f,l) \tag{8}$$

where μ_f is the mean of the column f.

Then, the rows should be standardized by removing any dc offset and normalizing the variance as follows:

$$\mu_l = \frac{1}{F}\sum_{f=1}^{F} \hat{X}(f,l) \tag{9}$$

$$\chi_l = \sum_{f=1}^{F} \hat{X}^2(f,l) \tag{10}$$

$$\Gamma_l = \sqrt{\frac{(\chi_l - F\cdot\mu_l^2)}{(F-1)}} \tag{11}$$

$$\hat{X}(f,l) = \frac{\hat{X}(f,l) - \mu_l}{\Gamma_l} \tag{12}$$

where μ_l is the mean, χ_l is the energy of the NASE, and Γ_l is the standard deviation of the row l. In a further step, the columns are whitened, which means that they are linearly transformed to remove any linear correlations between the dimensions. Whitening can be performed via eigenvalue decomposition of the covariance matrix

$$C = VDV^T = \mathrm{E}\left\{\hat{X}\hat{X}^T\right\} \tag{13}$$

$$C_P = D^{-(1/2)}V^T \tag{14}$$

where V is the matrix of orthogonal eigenvectors and D is a diagonal matrix with the corresponding eigenvalues. In order to perform dimensionality reduction, we reduce the size of the matrix C_P by throwing away $F - E$ of the columns of C_P corresponding to the smallest eigenvalues of D. We call the resulting matrix C_E, which has the dimensions $F \times E$. The whitening is done by multiplying the $F \times E$ transformation matrix C_E with the $L \times F$ matrix $\hat{X}$ as follows:

$$\check{X} = \hat{X}C_E. \tag{15}$$

This method of whitening is closely related to PCA. After extracting the reduced PCA basis C_E, a further step consisting of basis rotation in the directions of maximal statistical independence is needed for applications that require maximum decorrelation of features, such as the separation of source components of a spectrogram. A statistically independent basis is derived using an additional step of ICA after PCA extraction. The input $\check{X}$ is then fed to the FastICA algorithm, which maximizes the information in the following six steps.

Step 1) Initialize spectrum basis W_i to small random values, where i is the number of independent components.

Step 2) Apply Newton's method as follows:

$$W_i = \mathrm{E}\left\{\check{X} g\left(W_i^T \check{X}\right)\right\} - \mathrm{E}\left\{g'\left(W_i^T \check{X}\right)\right\} W_i \tag{16}$$

where g is the derivative of the nonquadratic function.

Step 3) Normalize the spectrum basis approximation W_i as follows:

$$W_i = \frac{W_i}{\|W_i\|}. \tag{17}$$

Step 4) Decorrelate using Gram–Schmidt orthogonalization as follows:

$$W_i = W_i - \sum_{j=1}^{i-1} W_i^T W_j W_j. \tag{18}$$

After every iteration step, subtract from W_i the projections $W_i^T W_j W_j$, $j = 1, \ldots, i$, of the previously estimated i vectors.

Step 5) Renormalize the spectrum basis approximation as follows:

$$W_i = \frac{W_i}{\|W_i\|} \tag{19}$$

Step 6) If not converged, go back to step 2).

The purpose of the Gram–Schmidt decorrelation/orthogonalization performed in the algorithm is to avoid finding the same component more than once. When the tolerance becomes close to zero, the Newton method will usually keep converging toward that solution, and so, by turning off the decorrelation when almost converged, the orthogonality constraint is loosened. Steps 1)–6) are executed until convergence. Then the iteration performing only the Newton step and normalization are carried out until convergence $W_i W_i^T = 1$. With this modification, the true maximum is found. The basis function $C_E W$ obtained by PCA and ICA is stored in the MPEG-7 basis function database for the classification scheme. The resulting spectrum projection is the product of the NASE matrix X, the dimension-reduced PCA basis functions C_E, and the ICA transformation matrix W as follows:

$$Y = X C_E W. \tag{20}$$

This spectrum projection is compliant with the spectrum projection from the MPEG-7 standard and is used to represent low-dimensional features of a spectrum after projection onto a reduced-rank basis.

B. Training HMMs

In order to train a statistical model on the basis projection features for each audio class, the MPEG-7 audio classification tool uses HMMs, which consist of several states. During training, the parameters for each state of an audio model are estimated by analysing the feature vectors of the training set. Each state represents a similarly behaving portion of an observable symbol sequence process. At each instant in time, the observable symbol in each sequence either stays at the same state or moves to another state depending on a set of state transition probabilities. Different state transitions may be more important for modeling different kinds of data. Thus, HMM topologies are used to describe how the states are connected. In television broadcasts, temporal structures of video sequences require the use of an ergodic topology, where each state can be reached from any other state and can be revisited after leaving. In our case, a five-state left–right model is suitable for speaker and isolated sound recognition. A left–right HMM with five states is trained for each sound class.

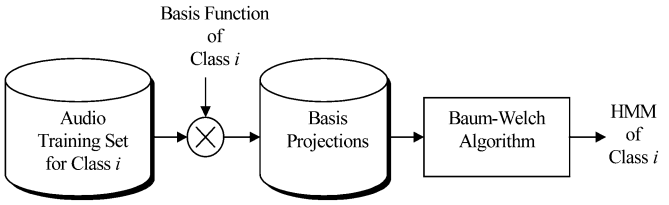

Fig. 3. HMM for a given sound class i.

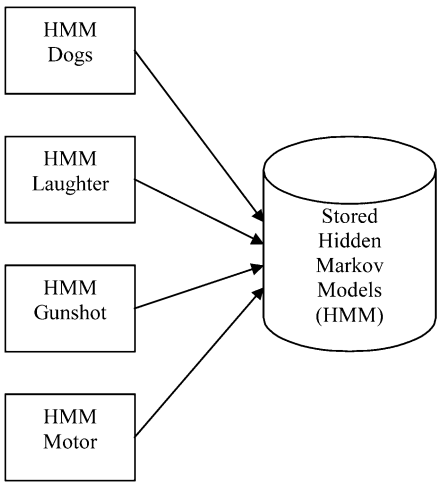

Fig. 4. Example classification scheme using HMMs.

Fig. 3 illustrates the training process of a HMM for a given sound class i.

The training audio data is first projected onto the basis function corresponding to sound class i. The HMM parameters are then obtained using the well-known Baum–Welch algorithm [14]. The procedure starts with random initial values for all of the parameters and optimizes the parameters by iterative reestimation. Each iteration runs through the entire set of training data in a process that is repeated until the model converges to satisfactory values. The parameters converged after three training iterations.

With the Baum–Welch reestimation training patterns, one HMM is computed for each class of sound that captures the statistically most regular features of the sound feature space. Fig. 4 shows an example classification scheme consisting of dogs, laughter, gunshots and motors. Each of the resulting HMMs is stored in the MPEG-7 sound classifier.

C. Sound Classification Using Spectrum Projections and HMMs

Sounds are modeled according to category labels and represented by a set of HMM parameters. Automatic classification of audio uses a collection of HMMs, category labels, and basis

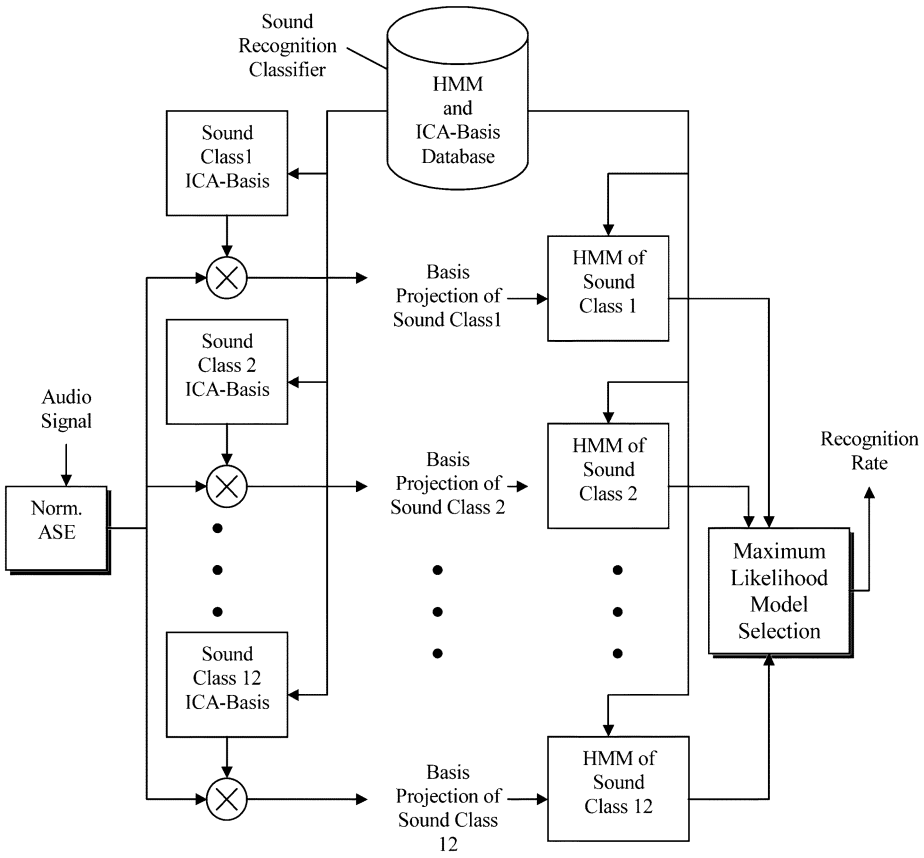

Fig. 5. Block diagram of the classification using spectrum basis projection features.

functions. Automatic audio classification finds the best-match class for an input sound by presenting it to a number of HMMs and selecting the model with the maximum likelihood score.

Here, the Viterbi algorithm is used as the dynamic programming algorithm applied to the HMM for computing the most likely state sequence for each model in the classifier given a test sound pattern. Thus, given a sound model and a test sound pattern, a maximum accumulative probability can be recursively computed at every time frame according to the Viterbi algorithm.

Fig. 5 depicts the recognition module used to classify an audio input based on pretrained sound class models (HMMs). Sounds are read from a media source format, such as WAV files. Given an input sound, the NASE features are extracted and projected against each individual sound model's set of basis functions, producing a low-dimensional feature representation. Then, the Viterbi algorithm is applied to align each projection on its corresponding sound class HMM (each HMM has its own representation space). The HMM yielding the best maximum-likelihood score is selected, and the corresponding optimal state path is stored.

D. Audio Retrieval Using a Histogram Sum of Squared Differences

An input sound is indexed by selecting the HMM yielding the maximum-likelihood score and storing the corresponding optimal HMM state path, which was obtained using the Viterbi algorithm. This state path describes the evolution of a sound through time with a sequence of integer state indices.

The MPEG-7 standard proposes a method for computing the similarity between two state paths generated by the Viterbi algorithm. This method, based on the sum of squared differences between "state path histograms," is explained in the following.

A normalized histogram can be generated from the state path obtained at the end of the classification procedure. Frequencies are normalized to values in the range $[0-1]$ obtained by dividing the number of samples associated with each state of the HMM by the total number of samples in the state sequence as follows:

$$hist_a(j) = \frac{N(j)}{\sum_{i=1}^{K} N(i)}, \quad 1 \le j \le K \tag{21}$$

where K is the number of states in the HMM and $N(j)$ is the number of samples for state j in the given state path.

A similarity measure between two state paths a and b is computed as the absolute difference between each relative frequencies summed over state indices $k\,(1 \le k \le K)$. This gives the Euclidian distance between the two sounds indexed by a and b as

$$\delta(a,b) = \sum_{j=1}^{k} \sqrt{(hist_a(j) - hist_b(j))^2}. \tag{22}$$

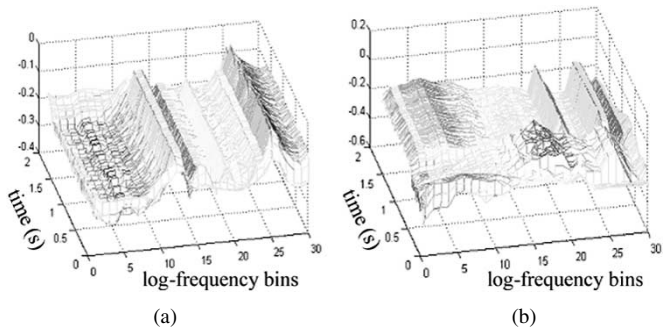

Fig. 6. NASE for: (a) an automobile horn and (b) for an old telephone ringing.

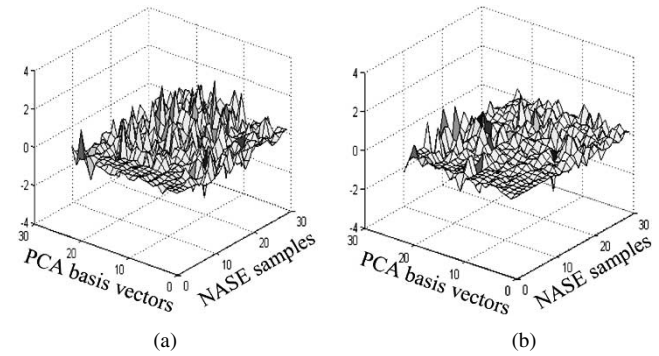

Fig. 7. PCA basis vectors: (a) for horns and (b) for a telephone ringing.

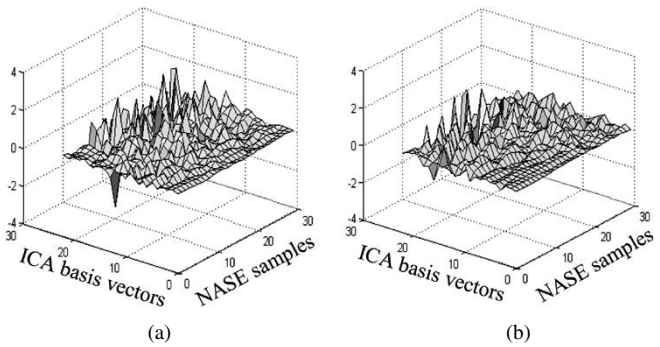

Fig. 8. FastICA basis vectors: (a) for horns and (b) for a telephone ringing.

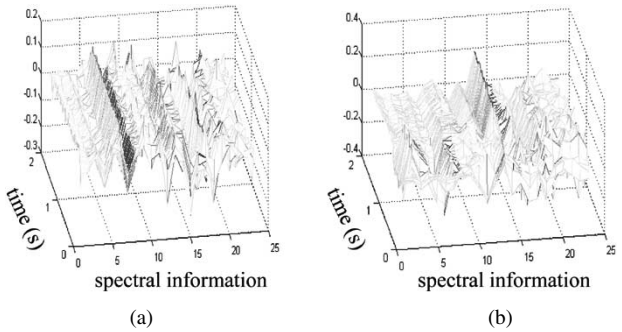

Fig. 9. Projection of NASE onto basis vectors for: (a) an automobile horn and (b) for an old telephone ringing.

III. Experimental Results

In order to evaluate the efficiency of our MPEG-7 audio descriptors, the reduced-dimension basis projection features were applied to a generalized sound and speaker classification system. The sound classification will be useful for film/video indexing, searching, and professional sound archiving. On the other hand, the speaker classification is useful for radio and television broadcasts.

A. Plots of MPEG-7 Audio Descriptors

To help the reader visualize the kind of information that the MPEG-7 audio descriptors convey, several results for four of the ASE and ASP descriptors are depicted in Figs. 6–9.

First we calculated the NASE, which is simply a power spectrum with logarithmically spaced frequency coefficients. The first coefficient represents power between 0 and the default "low edge" of 62.5 Hz, the next 28 coefficients represent 1/4-octave bands between 62.5 Hz and 8 kHz (seven octaves), and the 30th coefficient corresponds to the power between the "high edge" of 8 kHz and the Nyquist rate, which is 11.025 kHz.

The first sound is that of a typical automobile horn being honked once for about 1.5 s. Then the sound decays for roughly 200 ms. The reader should note that the harmonic nature of the honk, shown by the almost time=independent spectral peaks of the NASE $X(f,l)$, is readily visible in Fig. 6(a).

The decay at the end can also be seen as the higher frequencies decay and the lower frequencies seem to grow in strength. The lower frequencies becoming stronger may seem out of place, but this phenomenon is actually due to the normalization. As the sound in general becomes quieter, the levels at the different frequencies become more even and all are boosted by the normalization, even the low ones.

The NASE $X(f,l)$ of an old telephone being rung once is depicted in Fig. 6(a). The first 0.7 s consist of the noise-like sound of the manual cranking necessary for old-fashioned telephones, while the rest of the sound consists of the harmonic sound of the bells ringing out. That is, distinguishing between the harmonic and noise-like parts of sounds is easy per visual inspection of the NASE.

While the visual interpretation of the NASE is rather easy, visual interpretation of the bases C_E in Fig. 7 is not as straightforward.

Each of these bases is a matrix, which can be thought as a linear transformation between a spectral domain containing correlated information (NASE) and PCA basis vectors, in which the correlations in the information are reduced.

However, since we may not know exactly how the correlations are being reduced in each case, the bases are difficult to interpret. For instance, one can see in the PCA bases that the first basis vectors calculated are rather simple and have small variances, while the last basis vectors that are calculated tend to be complicated, have larger variances, and be less well behaved in general. This phenomenon corresponds to the fact that, as the algorithm extracts basis vectors, it becomes more and more difficult to find meaningful basis vectors because much of the information has already been extracted. The PCA algorithm also tends to find basis vectors that have large amplitudes, but not necessarily those that convey more information.

The FastICA algorithm, however, uses a nonlinear technique to help decorrelate the NASE. As a result, the bases generated via FastICA have more peaks on average due to larger variances.

The FastICA basis $C_E W$ is shown in Fig. 8(a) for horns and in Fig. 8(b) for telephone sounds.

The projections $Y = XC_E W$, on the other hand, look like versions of the NASE where the frequency information is scrambled by the basis. As can be verified in Fig. 9, telling apart the harmonic and noise-like parts of the sounds is still possible.

B. Experiments With Speaker Recognition

1) Datasets: For speaker recognition, 25 speakers were used, 11 male and 14 female. Each speaker was instructed to read 15 different sentences. After we used a sampling rate of 22.05 kHz to record the speakers reading the sentences, we cut the recordings into smaller clips: 16 training clips (about 60 s total), 5 additional longer training clips (60 s), and 5 test clips (20 s) per speaker. In order to determine if the amount of training data plays an important role for the different feature extraction methods, we defined two different training sets: the smaller set included only the 16 training clips and was 60 s long, and the larger set included the original 16 plus the 5 additional longer clips and was about 120 s long.

2) Classification and Results: Our goal was to compare the performance of NASE, PCA, ICA, and MFCC methods for speaker recognition.

For classification purposes, left–right HMM classifiers with five states were used to model each speaker. For each feature space (NASE, PCA, ICA, and MFCC), a set of 25 HMMs was trained using a classical expectation and maximization (EM) algorithm.

In the case of NASE, the matching process was easy because there were no bases. We simply matched each test clip against each of the 25 HMMs (trained with NASE features) via the Viterbi algorithm. The HMM yielding the best acoustic score (along the most probable state path) determined the recognized speaker.

In the case of the PCA and ICA methods, each HMM had been trained with data projected onto a basis as depicted in Fig. 5. So, every time we tested a sound clip on an HMM, we had to first project the sound clip's NASE onto the basis (ASB). This process caused testing to last considerably longer, as each test clip had to be projected onto 25 different bases, before it could be tested on the 25 HMMs to determine what it should be recognized as. On the other hand, the performance due to the projection onto the well-chosen bases increased recognition performance considerably. In order to perform a tradeoff between dimensionality reduction and information content maximization, feature extraction parameters in PCA and ICA needed to be selected with care.

The parameter with the most drastic impact turned out to be the horizontal dimension E of the matrix C_E from PCA. If E was too small, the matrix C_E reduced the data too much, and the HMMs did not receive enough information. However, if E became too large, then the extra information extracted was not very important and would have better been ignored. The recognition rate versus E from the PCA and ICA methods for the smaller training set are depicted in Fig. 10.

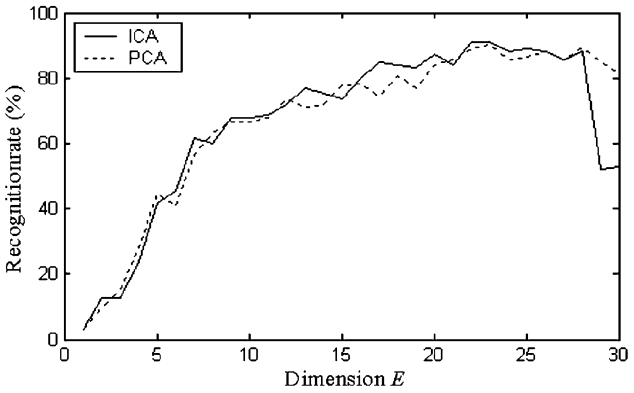

Fig. 10. Effect of E on recognition rates obtained with PCA and ICA.

TABLE I
SPEAKER RECOGNITION RESULTS

Recognition Mode	NASE	PCA	ICA	MFCC+delta +double delta
Speaker recognition (small set)	80.8	90.4	91.2	96.0
Speaker recognition (larger set)	80	85.6	93.6	98.4
Gender recognition (small set)	98.4	100	100	100

As can be seen above, the best value for both methods E was 23. However, this was not always the case. We also generated the plot for speaker recognition among six male speakers, which revealed that the optimal dimension for E should be 16, so it seems that one needs to be careful about choosing E and might have to test empirically to find the optimal value.

The results of our tests using the different feature extraction methods are shown in Table I.

For PCA and ICA, we simply took the recognition rate corresponding to $E = 23$, even though in one case the recognition rate was 1.5% higher for $E = 28$ (PCA, with larger training set).

Regarding the recognition of 25 speakers, ICA yields better performance than do PCA and NASE features. The resulting 93.6% recognition rate using ASE and ASP of MPEG-7 audio descriptors appears to be slightly higher than the 93.1% recognition rate of only 13 MFCCs, but explicitly lower than the 98% recognition rate that we obtained with 13 MFCCs, their 13 delta and 13 double-delta acceleration coefficients because dynamic features such as delta and double-delta provide estimates of a gross shape (linear and second-order curvature) of a short segment of feature trajectory. It appears that MFCC, which is not an MPEG-7 feature, outperforms MPEG-7. To test gender recognition, we used the smaller set. Two HMMs were trained: one with the training clips from female speakers and the other with the training clips from male speakers. Because there were only two possible answers to the recognition question: male or female, this experiment was naturally much easier to carry out and resulted in excellent recognition rates, as depicted in Table I. The

term 100% indicates that zero mistakes were made out of 125 test sound clips.

C. Sound Recognition

1) Building the Sound Libraries: To test the sound recognition system, we built sound libraries from various sources including the speech database that we collected for speaker recognition (see Section III-B1) and the "Sound Ideas" general sound effects library. We created 12 sound classes containing 40 training and 20 different testing sound clips, which were recorded at 22 kHz and 16 b and which ranged from 1 to 3 s long.

2) Classification Using a One-Level Structure: We used a simple sound recognition system using a one-level structure (no hierarchy), as depicted in Fig. 11.

After calculating the NASE for each of our training clips, we used this data to calculate a basis for each class using the FastICA algorithm. Then, we projected the NASEs from the training clips onto their respective bases and used these projections to train one HMM per class.

For testing, we calculated the NASE for each test clip and projected this data onto all of the bases generated from the training data. Next, we passed the projections to their respective hidden Markov models to calculate the maximum-likelihood scores. The highest score was used to determine the test clip's recognized sound class.

It is important to note that, for the recognition results, we did not use any sort of hierarchy to find a path from a root node to the recognized sound class; rather, all of the classes were tested at once and compared on one level.

This method is the most straightforward but would cause problems when there were too many classes on the one level.

3) Classification Using a Hierarchical Classification Structure: We organized the database of sound classes on the hard disk using the hierarchy shown in Fig. 12, assuming that particular sound classes, such as *female speech* and *male speech*, were more closely related than others such as *female speech* and *gun*.

Because we had modeled the database in this fashion, we decided to try using the same hierarchy for recognition, to see what effect it would have on the recognition rate. That is, we created additional bases and HMMs for the more general classes *animal*, *foley*, *people*, and *speech*.

For each test sound, a path was found from the root down to a leaf node with testing occurring at each level in the hierarchy.

4) Classification Using a Hierarchical Classification Structure With Hints: In certain systems, it would be feasible to assume that additional information were already available. For instance, it would be possible to have a recording of human speech but not be able to tell the gender of the speaker by ear.

The hint *speech* could be given, so that the program could determine what gender the speaker is with higher accuracy. In the following, each sound clip was assigned a hint, so that only one decision per clip needed to be made by the sound recognition program.

5) Results With the Different Classification Methods: Table II describes the recognition results with different classification structures.

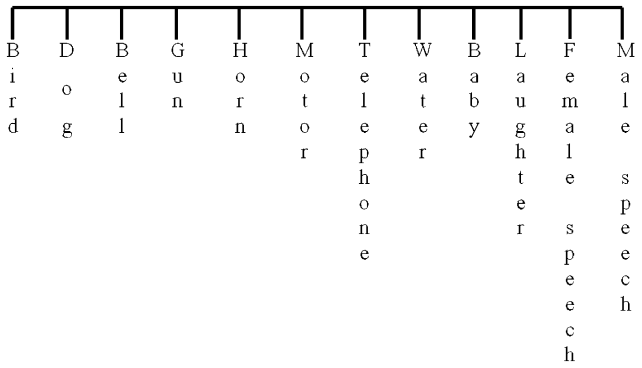

Fig. 11. Classification using a one-level structure (i.e., no hierarchy). Compare with Fig. 12.

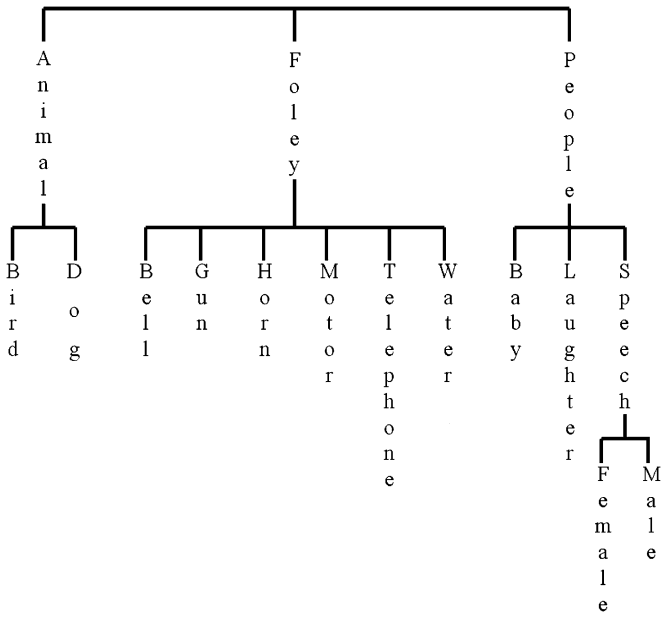

Fig. 12. Hierarchy for classification using a tree structure.

TABLE II
RECOGNITION RATES OF DIFFERENT COMPARISON STRUCTURES

Methods	Best Recognition Rate (%)	Value of E
One level classification (no hierarchy)	96	21
Hierarchical classification without hints	91	25
Hierarchical classification with hints	99	23

We achieved a 96% recognition rate with the classification using a single-level structure. This recognition rate appears to be slightly lower than the 97.7% recognition rate obtained with 39 MFCCs (13 cepstral coefficients plus their first- and second-order derivatives).

In the classification using hierarchical classification without hints, the best recognition rate that we achieved was 91% with the dimension $E = 25$. The recognition rate is lower compared to that of the single-level structure because the system could

TABLE III
RESULTS USING A TREE STRUCTURE

Similar sound	Maximum likelihood score	Euclidean distance
Telephone 37	37.8924	0.111033
Telephone 34	38.5650	0.111627
Telephone 58	38.4153	0.116466
Telephone 35	25.3898	0.135812
Telephone 55	39.2438	0.150099
Telephone 60	36.1829	0.158053

not handle the generality well. Many of the new errors were due to problems with recognition in the highest layer, which we attribute to the fact that sound samples in different branches of the tree were too similar. For example, some bird sounds and horn sounds were difficult to tell apart with the human ear. Using a hierarchical structure for sound recognition does not necessarily improve recognition rates if sounds in different general classes are too similar unless some sort of additional information (e.g., a hint) is available. The hierarchical classification with hints yields a higher recognition rate than a one-level structure or hierarchical classification without hints. We obtained a recognition rate of 99% via the FastICA algorithm with the optimal dimension $E = 23$. Again, the general shape of the plot of the recognition rates was the same as in Fig. 10.

6) Audio Retrieval Results: Once an input sound a has been recognized as a sound of class Cl, the state paths of the sounds b in the MPEG-7 database, which belong to class Cl, can be compared to the state path of a using the Euclidean distance $\delta(a, b)$ as described in Section II-D. These sounds can then be sorted so that those corresponding to the smallest distances are at the top of the list. That is, the items which are the most similar to the query should be at the top of the list and the most dissimilar ones at the bottom.

This system would basically be a search engine for similar sounds within a given sound class. In the example below, telephone_28 was input as a test sound a and recognized as $Cl = telephone$. The list of the retrieved items indexed with *telephone*, sorted by similarity with query telephone_28, is shown in Table III.

The maximum-likelihood scores used for classification are also included in Table III, so that the reader can note that calculating the similarity by comparing the state paths and by comparing the maximum-likelihood scores produce different results. As far as we know, there have not been any tests to show which technique of calculating similarity better corresponds to that of the human hearing system.

If, however, the sound a had been incorrectly recognized as something else, such as *bell*, we would have searched in the *bell* class for similar training data and found irrelevant results. Thus, it would have been better to have also searched the second and/or third most likely classes while hoping that *telephone* were one of these.

In the tests of this method, we used training data from the three classes with the highest maximum-likelihood scores to produce a list sorted by the Euclidean distance between the query a and each of the retrieved items. This method seemed promising but would often produce inconsistent lists including data from different classes mixed up with each other. We decided to compare the reliability of this retrieval technique using different methods for recognition.

TABLE IV
CONSISTENCIES

Method	With the state paths	With maximum likelihood score
NASE	0.69	0.50
PCA	0.72	0.57
FastICA	0.73	0.58

To compare lists of similar items, we used our own measure called *consistency*. A list is consistent when the elements next to each other belong to the same class, and a list is inconsistent when any two adjacent elements always belong to different classes. We used the following method to calculate the consistency C of a retrieval method. M sound clips are tested to produce M lists l_m of similar sounds, such that $1 \leq m \leq M$. Let L_m be the length of the list l_m, and let N_m be the number of times that two adjacent entries in the list l_m belong to the same class. Compute the consistency C according to

$$C_m = \frac{N_m}{L_m - 1} \tag{23}$$

$$C = E\{C_M\} \cong \frac{1}{M} \sum_{m=1}^{M} C_m. \tag{24}$$

Thus, the consistency is a real number between 0 and 1, where 0 is as inconsistent as possible and 1 is as consistent as possible.

Using the same library of test sounds, we then measured the inconsistency for retrieval methods using NASE, PCA projections, and FastICA projections as input to the HMMs. As it was also possible to measure the similarity using just the maximum-likelihood scores, we also listed those results in Table IV.

The results reflect what we expected, namely, that the lists of similar sounds were more consistent, if we used the state paths instead of the maximum-likelihood scores for comparison. We attribute this result to the fact that the state paths contain more information because they are multidimensional whereas the maximum-likelihood scores are one-dimensional. Thus, our best technique for retrieving similar sounds is the FastICA method using the state paths for comparison.

IV. CONCLUSION

In this paper, we applied the ASB and ASP MPEG-7 audio descriptors to two recognition systems: a speaker recognizer and a sound classification and retrieval system. The speaker recognizer, tested with 14 female and 11 male speakers, yields a high recognition rate. For comparison, standard MFCC with delta and double-delta features were extracted. The experimental results showed that the recognition rate using 23 dimensional ASP features was slightly lower than 39 dimensional MFCC feature vectors. But the ASP features using ICA basis functions demonstrated better speaker and gender recognition performance than the NASE features and the PCA basis projection features.

The sound classification module achieved high recognition rates on 12 different sound classes. The use of a hierarchical

structure with hints of sound classes improved the classification accuracy compared to a hierarchical structure without hints and a single-level system.

The sound retrieval system relies on the performance of the classifier. Our retrieval system is based on a distance metric allowing to compare 2 state paths in conformance with the audio part of the MPEG-7 standard. The approach proposed in this paper consists in retaining the three most likely sound class hypotheses in order to cope with possible recognition errors. Even in case of a classification error, most of the retrieved sounds may thus be part of the correct class. We also proposed a consistency measure to evaluate the homogeneity of such mixed result lists. Some tests reveal that PCA- and ICA-based classification yield more consistent retrieval results using MPEG-7 conform state path descriptors rather than HMM acoustic scores. This retrieval approach should be refined in the future by investigating the use of additional MPEG-7 features for training and testing along with other methods for evaluating the consistency of a retrieval result list. Moreover, it would be interesting to compare the retrieval results with human subjective tests of sound similarity. In future research, we will focus on improving the MPEG-7 sound classification using various HMM topologies. This will be applied to news and home video hierarchical indexing.

REFERENCES

[1] S. Davis and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 357–366, Aug. 1980.

[2] J. Foote, "Content-based retrieval of music and audio," *Multimed. Storage Archiv. Syst. II*, pp. 138–147, Aug. 1997.

[3] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, pp. 27–36, Fall 1996.

[4] G. Guo, H. Zhang, and S. Z. Li, "Boosting for content-based audio classification and retrieval: an evaluation," Microsoft Research Tech. Rep. MSR-TR-2001-15.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comp. Syst. Sct.*, vol. 55, no. 1, pp. 119–139, 1997.

[6] S. Z. Li and G. Guo, "Content-based audio classification and retrieval using SVM learning," in *Invited Talk PCM*, 2000.

[7] J. Thorsten, *Learning to Classify Text Using Support Vector Machine*. Boston, MA: Kluwer, 2002.

[8] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[9] M. A. Casey, "General sound similarity and sound recognition tools," in *Introduction to MPEG-7*, B. S. Manjunath, P. Salembier, and T. Sikora, Eds. New York: Wiley, 2000.

[10] ——, "MPEG-7 sound recognition tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, May/June 2001.

[11] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7*. New York: Wiley, 2000.

[12] *Information Technology Multimedia Content Description Interface-Part 4: Audio*, ISO/IEC JTC 1/SC 29, June 2001.

[13] P. Comon, "Independent component analysis, a new concept?," *Neural Computation*, vol. 7, no. 6, pp. 1004–1034, 1995.

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

[15] I. T. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer-Verlag, 1986.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1993.

[17] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, Eds. Norwell, MA: Kluwer, 2003, pp. 91–109.

[18] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.

Hyoung-Gook Kim received the diploma degree in electrical engineering and the Ph.D. degree in computer science from the Technical University of Berlin, Berlin, Germany, in 1997 and 2002, respectively.

From 1998 to 1999, he worked on mobile service robots at Daimler Benz AG and speech recognition at Siemens AG. From 1999 to 2002, he was the Project Leader of the Speech Processing Laboratory at Cortologic AG, where he developed a noise reduction preprocessor and a 1.2-kb/s low-bit-rate speech coder for mobile voice communication. In August 2002, he joined the Communication Systems Department, Technical University of Berlin, where he currently is a Senior Researcher on the MPEG-7 Annotation of Video Sequences project. His current research interests include audio analysis, indexing and classification, automatic segmentation, and distributed speech recognition system. He has published extensively in the field of audio and speech processing.

Dr. Kim is an active member of ISO MPEG.

Nicolas Moreau received the Engineer degree from the Ecole Nationale Supérieure de Télécommunication de Bretagne, Bretagne, France, in 1997 and the Ph.D. degree in computer science and signal processing from the University of Rennes 1, Rennes, France, in 2001.

His Ph.D. work was carried out at the France Telecom R&D Centre, Lannion, France, within the Speech Recognition Laboratory. Since 2002, he has been a Senior Researcher working on the MPEG-7 Annotation of Video Sequences project of the Communication Systems Department, Technical University of Berlin, Berlin, Germany. His main fields of research are automatic speech recognition, spoken document retrieval, and the spoken content description part of the MPEG-7 standard. He has published several papers in the field of speech processing.

Dr. Moreau is an active member of ISO MPEG.

Thomas Sikora (M'93–SM'96) received the Dipl.-Ing. degree and Dr.-Ing. degree in electrical engineering from Bremen University, Bremen, Germany, in 1985 and 1989, respectively.

He is a Professor and Director of the Communication Systems Department, Technical University of Berlin, Berlin, Germany. In 1990 he joined Siemens Ltd. and Monash University, Melbourne, Australia, as a Project Leader responsible for video compression research activities in the Australian Universal Broadband Video Codec consortium. He became a Member of the Research Staff of the Heinrich-Hertz-Institute (HHI), Berlin, in 1994 and directed the Interactive Media Department at HHI between 1997 and 2001. He has been involved in international ITU and ISO standardization activities as well as in several European research activities for a number of years. He acted as the chairman of the ISO-MPEG video group (Moving Picture Experts Group) between 1995 and 2001, responsible for the development and standardization of the MPEG-4 and MPEG-7 video coding algorithms. He also served as the chairman of the European COST 211ter video compression research group. He frequently works as an industry consultant on issues related to interactive digital audio and video. He is an appointed member of the Advisory and Supervisory board of a number of German companies and international research organizations. He has published one book and more than 200 refereed journal and conference papers in the field of image, video and audio processing, and he has been an invited plenary speaker at a number of international conferences.

Dr. Sikora is a recipient of the 1996 German ITG award (German Society for Information Technology). He is the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is an Associate Editor of the *EURASIP Signal Processing* journal and an advisory editor for the *EURASIP Signal Processing: Image Communication* journal. From 1996 to 2000, he was on the Editorial Board the IEEE SIGNAL PROCESSING MAGAZINE. He is a member of ITG.