Musical Applications of MPEG-7 Audio

Michael Casey, MERL Cambridge Research Laboratory

Abstract— the MPEG-7 international standard contains new tools for computing similarity and classification of audio clips and for extracting prominent acoustic features from mixed audio scenes. Among the applications for the standardized tools are enhanced software tools for composition and sound editing. In this paper we present these tools and discuss possible directions for future applications.

Keywords: source separation, sound similarity, generalized timbre, sound classification.

I. Introduction

As databases of sound samples and music files become larger and more interconnected, composers and sound designers are faced with new challenges in content management. MPEG-7 is a new international standard for media content description that is well suited to applications in music indexing, similarity matching, and knowledge-based audio processing. Support for content management applications was a part of the design requirements for the standard and, as such, it consists of methods for computing feature extraction, similarity, and classification on a wide range of sound types. Included within the standard are methods for describing speech, vocal utterances, environmental sounds, musical sequences, silence and mixed auditory scenes. By introducing these methods in the public domain, MPEG-7 will likely have a similar scale of impact on the future of music technology as the MIDI and MPEG-2 layer III audio (MP3) standards have had in the past.

In this paper we shall discuss two new novel components of MPEG-7 audio; namely, independent subspace analysis (ISA) and sound similarity and classification using probabilistic inference models. These methods were chosen from a range of competing technologies and were found to exhibit good performance in a wide variety of applications.

Independent subspace analysis is a signal processing method for separating multiple independent layers of a spectrogram from a single-channel or stereo mixtures; (Casey and Westner 2000). These independent layers yield individual acoustic sources from a mixture thus enabling sound *unmixing* and *remixing* applications.

The generalized sound recognition tools use independent subspace features coupled with hidden Markov models (HMM) for representing similarity and source classifications for different sounds. In contrast to musical timbre similarity methods, the MPEG-7 sound similarity methods include sounds other than isolated musical instrument notes, such as noise textures, environmental sounds, melodies, utterances and mixtures of sources. For other work in this area see for example (Wold, Blum, Keislar, and Wheaton 1996; Boreczky and Wilcox 1998; Martin and Kim1998; Zhang and Kuo 1998). These tools will be useful to composers for automatically organizing sonic materials using computational methods. The tools will also be of interest to software developers and researchers for building new advanced applications for music and audio processing.

II. Independent Subspace Analysis and Spectrum Basis Functions

Figue 1 shows an application called *SoundSplitter* that analyzes an input spectrogram consisting of a mixture of sources and produces a number of output spectrograms corresponding to the individual sources. On the right hand side of the figure a series of time and frequency masking functions are shown. These functions are basis functions that describe the statistically most salient features in the mixed spectrum in terms of time and frequency profiles. The profiles are used to estimate the individual source output spectrograms shown below the input spectrogram in the figure. These basis functions for a spectrum are estimated using a new audio analysis technique call independent subspace analysis (ISA); see (Casey and Westner 2000).

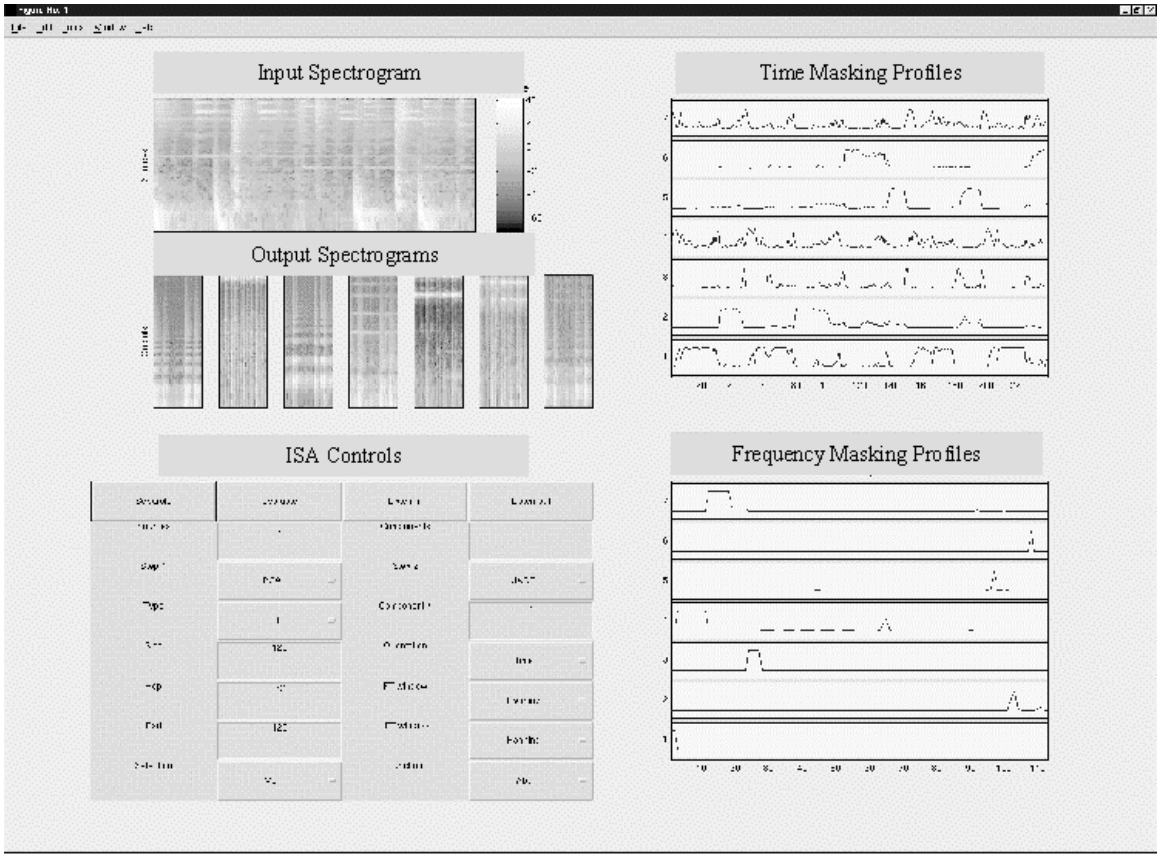

Figure 1. The *SoundSplitter* application for independent subspace analysis of audio.

The technique of independent subspace analysis (ISA) was developed to describe individual source components within a single-channel mixture. The technique generates basis functions for a spectrogram that are used to compute spectral masking functions. The time masking functions show the individual rhythm envelopes and the frequency masking functions show groups of frequency components that are correlated in the mixed spectrum. In general, correlated components across frequency indicate a consistent underlying source within the input spectrogram. For audio clips that contain time-varying sources, a larger number of basis functions is required than in the example shown. In this case, the time-varying audio segment must be broken into sub-segments of between 0.25s and 1.0s with each sub-segment's spectrogram decomposed into basis functions as in the drum mixture example.

Figure 2 shows a close up view of the input spectrogram from Figure 1. The individual estimated source spectrograms corresponding to the bass drum, snare drum and cow bells are shown in Figures 3-5. The source spectrograms are reconstructed using independent subspace synthesis, a technique that re-filters the source spectrogram using the estimated time and frequency masking functions. The re-filtered spectrogram may be further processed using the inverse Fourier transform thus yielding a separated audio signal, see Figure 6. The re-filtering operation substitutes phases from the original source spectrogram but masks the magnitude spectrum using the masking profiles of the ISA functions. This basis function representation is a practical technique that may be used for sample manipulation and sound editing. It is also useful for low-dimensional quantitative description of a spectrum and may be efficiently applied to the problem of sound source recognition as well as separation.

The ISA representation of audio segments is contained in MPEG-7 as part of the *low-level audio descriptors* (LLD) tool suite. There are many such tools within the standard, many are familiar tools such as harmonic spectral peak descriptions and fundamental frequency. However, we shall focus on the novel approaches to audio description that have been adopted by the MPEG-7 standard.

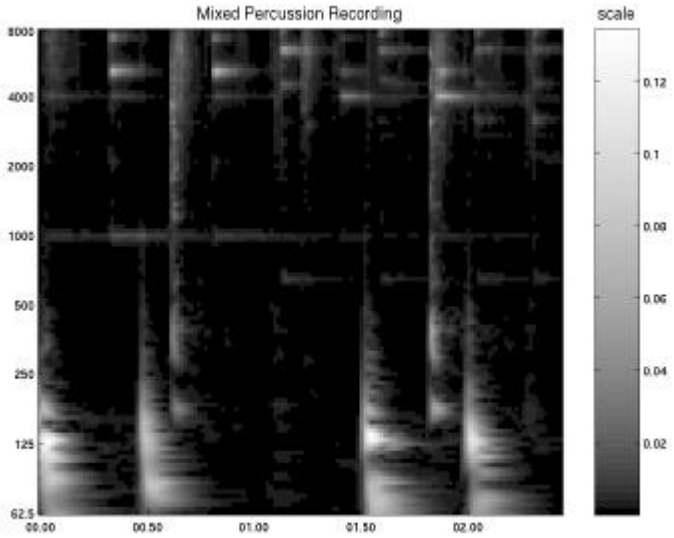

Figure 2. Log-frequency power spectrum of a mixed percussion recording.

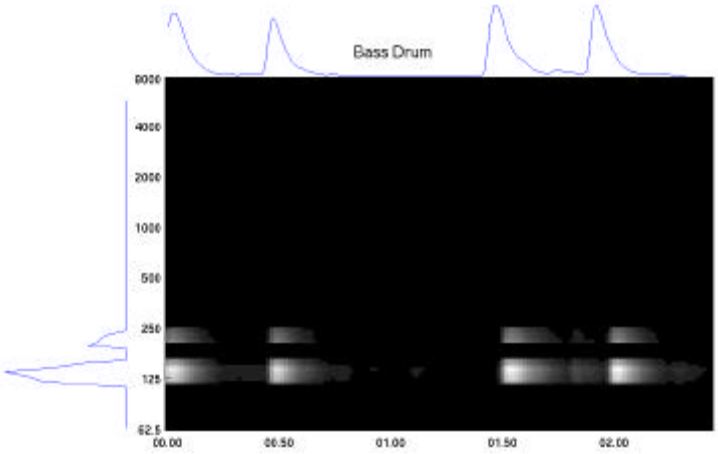

Figure 3. Spectrogram reconstruction of the bass drum estimated by re-filtering the input spectrogram using ISA basis functions.The function to the left is frequency mask component and the function across the top is the time masking component.

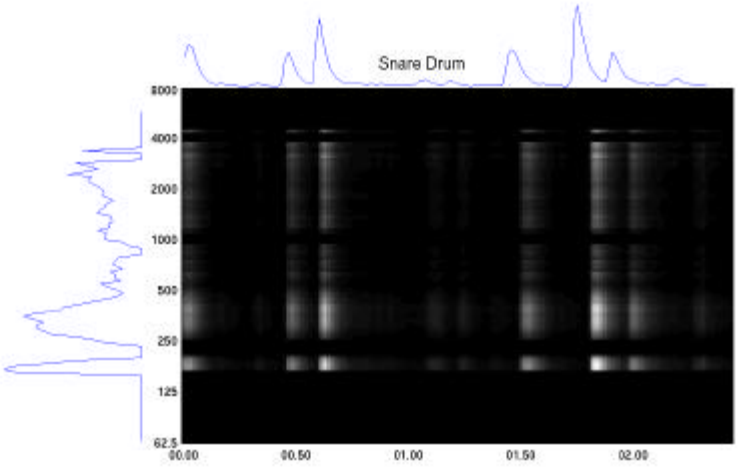

Figure 4. Masking functions and spectrogram reconstruction of the snare drum.

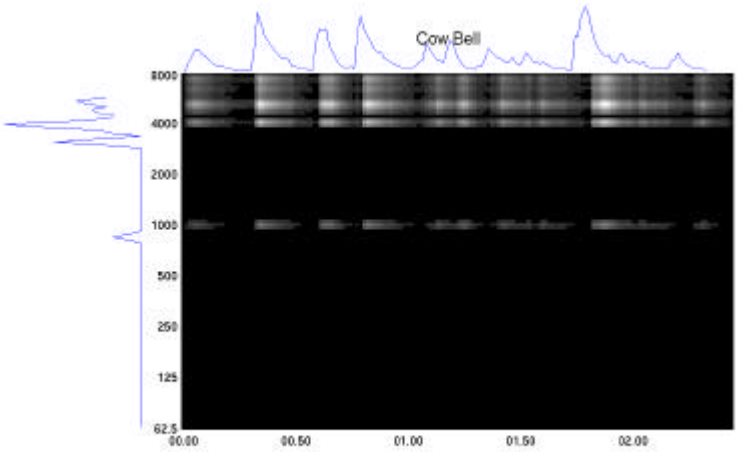

Figure 5. Masking functions and spectrogram reconstruction of the cow bell.

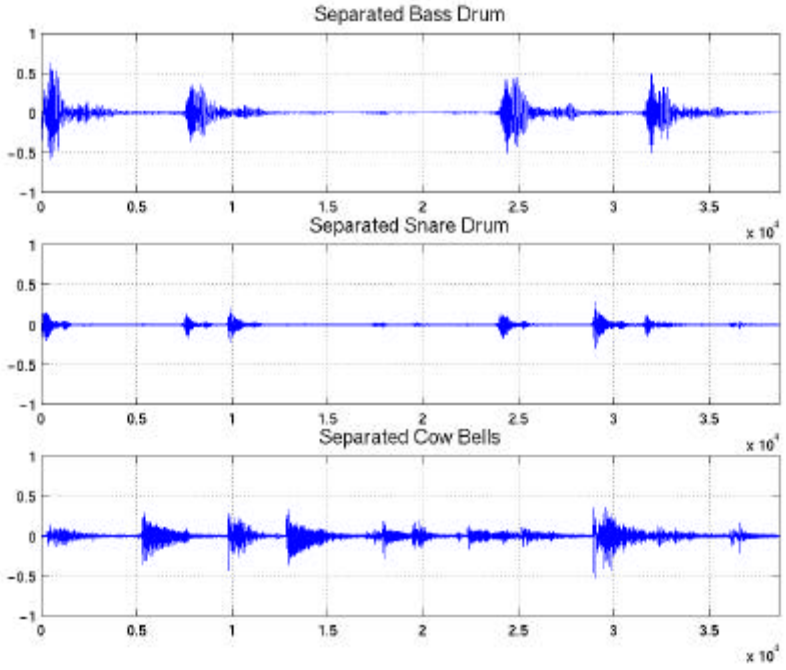

Figure 6. Separated audio signals using ISA basis functions with spectrogram re-filtering.

A. Independent Subspace Analysis within MPEG-7

The MPEG-7 standard consists of descriptors and description schemes that are defined by a modified version of XML schema called the MPEG-7 description definition language (DDL). A large number of descriptors have been defined covering images, audio, video and general multimedia usage. The DDL language ensures that media content description data may be shared between applications in much the same way that sound files are exchanged using standard file formats. For example, an audio spectrum is defined by a descriptor called `AudioSpectrumEnvelope`. To use the descriptor, data is instantiated using the standardized DDL syntax. In this case, the spectrum data is stored as a series of vectors within the class.

The `AudioSpectrumBasis` descriptor contains basis functions that are used to project high-dimensional spectrum descriptions into a low-dimensional representation contained by the `AudioSpectrumProjection` descriptor, see DDL Example 1. These two sets of functions correspond to the time functions and frequency functions of ISA analysis described above. The dimensionality of a spectrum is simply the number of channels of spectral data. In the example above, the representation was used for describing independent component spectrograms for source mixture separation. The reduced representation is also well suited for use with probability model classifiers that require input features to be of fewer than 10 dimensions for successful performance. The reduced dimension basis functions (time and frequency masks) behave as uncorrelated descriptions of the input spectrogram with the features described much more efficiently than using the full spectrogram data set. These features were found to exhibit superior performance in sound recognition tasks as we shall describe later.

```
<AudioD xsi:type="AudioSpectrumBasisType" loEdge="62.5" hiEdge="8000"
          resolution="1/4 octave">
    <BasisFunctions>
      <Matrix dim="10 5">
        0.26 -0.05 0.01 -0.70 0.44
        0.34 0.09 0.21 -0.42 -0.05
        0.33 0.15 0.24 -0.05 -0.39
        0.33 0.15 0.24 -0.05 -0.39
        0.27 0.13 0.16 0.24 -0.04
        0.27 0.13 0.16 0.24 -0.04
        0.23 0.13 0.09 0.27 0.24
        0.20 0.13 0.04 0.22 0.40
        0.17 0.11 0.01 0.14 0.37
        0.33 -0.15 0.24 0.05 0.39
```

```
      </Matrix>
   </BasisFunctions>
</AudioD>
```

DDL Example 1. Description of five basis functions using AudioSpectrumBasisType. The description definition language is based on XML schema with some extensions specific to MPEG-7. (The floating-point resolution has been truncated for clarity).

B. Independent Subspace Extraction Method

The extraction method for `AudioSpectrumBasis` and `AudioSpectrumProjection` is detailed within the MPEG-7 standard. It is considered that these steps *must* be used in extracting a reduced-dimension description in order to conform to the standard. Within each step there is opportunity for alternate implementations. As such, the following procedure outlines the standardized extraction method for ISA basis functions:

1. *Power spectrum*: instantiate an `AudioSpectrumEnvelope` descriptor using the extraction method defined in `AudioSpectrumEnvelopeType`. The resulting data will be a SeriesOfVectors with *M* frames and *N* frequency bins.

2 *Log-scale norming*: for each spectral vector, **x**, in `AudioSpectrumEnvelope`, convert the power spectrum to a decibel scale:

$$\mathbf{z} = 10\log_{10}(\mathbf{x})$$

and compute the *L2-norm* of the vector elements:

$$r = \sqrt{\sum_{k=1}^{N} z_k^2}$$

the new unit-norm spectral vector is calculated by:

$$\tilde{\mathbf{x}} = \frac{\mathbf{z}}{r}$$

3 *Observation matrix*: place each vector *row-wise* into a matrix. The size of the resulting matrix is *M x N* where *M* is the number of time frames and *N* is the number of frequency bins. The matrix will have the following structure:

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \tilde{\mathbf{x}}_2^T \\ \vdots \\ \vdots \\ \tilde{\mathbf{x}}_M^T \end{bmatrix}$$

4 *Basis extraction*: Extract a basis using a singular value decomposition (SVD); commonly implemented as a built-in function in many software packages using the command `[U,S,V] = SVD(X,0)`. Use the *economy* SVD when available since the row-basis functions are not required and this will increase extraction efficiency. The SVD factors the matrix from step 3 in the following way:

$$\tilde{\mathbf{X}} = \mathbf{USV}^T$$

where **X** is factored into the matrix product of three matrices; the row basis **U,** the diagonal singular value matrix **S** and the transposed column basis functions **V.** Reduce the basis by retaining only the first *K* basis functions, i.e. the first *K* columns of **V**:

$$\mathbf{V}_K = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_k \end{bmatrix}$$

K is typically in the range of 3-10 basis functions for feature-based applications. To calculate the proportion of information retained for *K* basis functions use the singular values contained in matrix **S**:

$$I(K) = \frac{\sum_{i=1}^{K} S(i,i)}{\sum_{j=1}^{N} S(j,j)}$$

where *I(K)* is the proportion of information retained for *K* basis functions and *N* is the total number of basis functions which is also equal to the number of spectral bins. The SVD basis functions are stored in the columns of a matrix within the `AudioSpectrumBasisType` descriptor.

6 *Statistically independent basis (Optional)*: after extracting the reduced SVD basis, **V,** a further step consisting of basis rotation to directions of maximal statistical independence is often desirable. This is necessary for displaying independent components of a spectrogram and for any application requiring maximum separation of features.

To find a statistically independent basis using the basis functions obtained in step 4, use one of the well-known, widely published independent component (ICA) algorithms such as INFOMAX, *JADE* or *FastICA;* (Bell and Sejnowski 1995; Cardoso and Laheld 1996; Hyvarinen, 1999).

The ICA basis is the same size as the SVD basis and is stored in the columns of the matrix contained in the `AudioSpectrumBasisType` descriptor. The retained information ratio, *I(K),* is equivalent to the SVD when using the given extraction method.

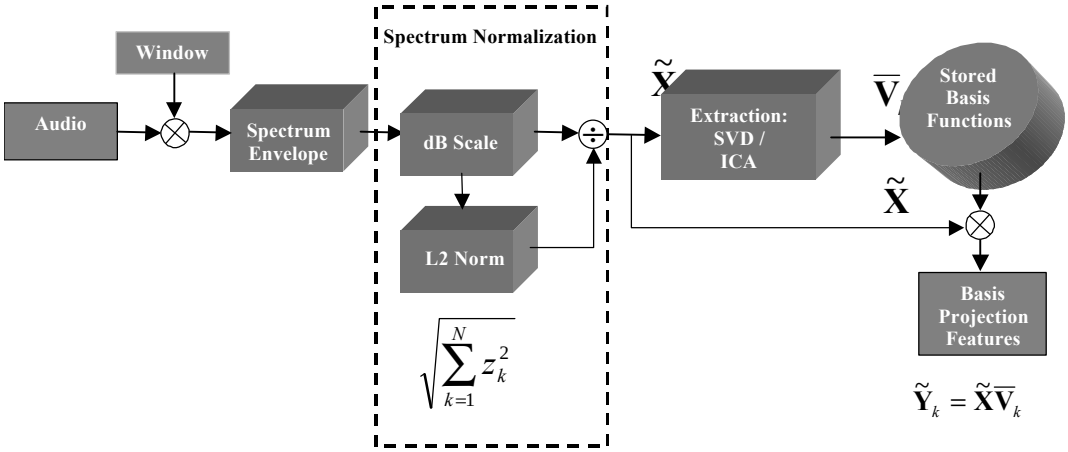

Figure 7 Extraction method for `AudioSpectrumBasisType` and `AudioSpectrumProjectionType`

C. Basis Projection Features (Time Mask Functions)

Figure 7. shows the extraction system diagram for both `AudioSpectrumBasis` and `AudioSpectrumProjection.` The basis projection gives time masking functions that are combined with the spectrum basis functions to reconstruct independent spectrogram components. To perform extraction for `SpectrumBasisProjection` follow steps 1-3 described above for `AudioSpectrumBasis` extraction, this produces a spectrum matrix. The only further requirement is to multiply the spectrum matrix with the basis vectors obtained in step 4 or, optionally, step 5. The method is the same for both SVD and ICA basis functions:

$$\tilde{\mathbf{Y}}_k = \tilde{\mathbf{X}}\overline{\mathbf{V}}_k$$

where **Y** is a matrix consisting of the reduced dimension features after projection of the spectrum against the basis **V**.
For independent spectrogram reconstruction, extract the non-normalized spectrum projection by skipping the normalization step (2) in `AudioSpectrumBasis` extraction. Thus:

$$\mathbf{Y}_k = \mathbf{X}\overline{\mathbf{V}}_k$$

Now, to reconstruct an indpendent spectrogram component use the individual vector pairs, corresponding to the *K*th vector in `AudioSpectrumBasis` and `AudioSpectrumProjection`, and apply the reconstruction equation:

$$\mathbf{X}_k = \mathbf{y}_k \overline{\mathbf{v}}_k^{+}$$

where the + operator indicates the transpose for SVD basis functions (which are orthonormal) or the pseudo-inverse for ICA basis functions (non-orthogonal).

The method outlined above represents a powerful tool that can be used for many purposes. The extracted sources may be subjected to further analysis such as tempo estimation, rhythm analysis or fundamental frequency extraction. For example, we now consider how ISA features may be used for sound recognition and similarity judgements for general audio.

III. GENERALIZED SOUND RECOGNITION

A number of tools exist within the MPEG-7 framework for computing similarity between segments of audio. In this section we describe tools for representing category concepts as well as tools for computing similarity in a general manner. The method involves training statistical models to learn to recognize the classes of sound defined in a taxonomy.

A. Taxonomies

A taxonomy consists of a number of sound categories organized into a hierarchical tree. For example, voice, instruments, environmental sounds, animals, etc. Each of these classes can be broken down further into more detailed descriptions such as: female laughter, rain, explosions, birds, dogs, etc.

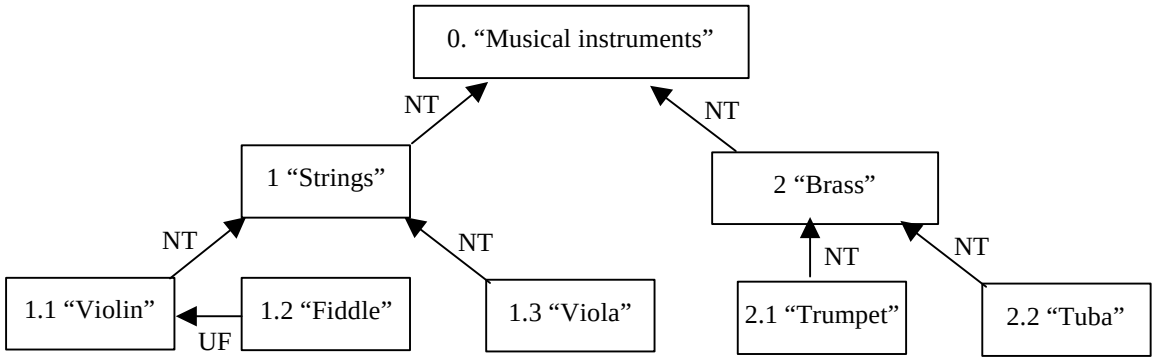

Figure 8. A controlled-term taxonomy of part of the *Musical Instruments* hierarchy

Figure 8 shows musical instrument controlled terms that are organized into a taxonomy with “Strings” and “Brass”. Each term has at least one relation link to another term. By default, a contained term is considered a narrower term (NT) than the containing term. However, in this example, “Fiddle” is defined as being a nearly synonymous with, but less preferable than, “Violin”. To capture such structure, the following relations are available as part of the `ControlledTerm` description scheme:

- *BT* – **Broader term**. The related term is more general in meaning than the containing term.
- *NT* – **Narrower term**. The related term is more specific in meaning than the containing term.
- *US* – **Use** The related term is (nearly) synonymous with the current term but the related term is preferred to the current term.
- *UF* – **Use for**. Use of the current term is preferred to the use of the (nearly) synonymous related term.

- *RT* – **Related Term**. Related term is not a synonym, quasi-synonym, broader or narrower term, but is associated with the containing term.

The purpose of the taxonomy is to provide semantic relationships between categories. As the taxonomy gets larger and more fully connected the utility of the category relationships increases. Figure 9 shows the taxonomy in Figure 8 combined into a larger classification scheme including animal sounds, musical instruments, Foley sounds (sound effects for film and television), and impact sounds. By descending the hierarchical tree we find that there are 17 leaf nodes in the taxonomy. By inference, a sound segment that is classified in one of the leaf nodes inherits the category label of its parent node in the taxonomy. For example, a sound classified as a "Dog Bark" also inherits the label "Animals". We shall adhere to this taxonomy for illustrative purposes only; MPEG-7 allows full flexibility in defining taxonomies using controlled terms and can be used to define much larger taxonomies than the given example.

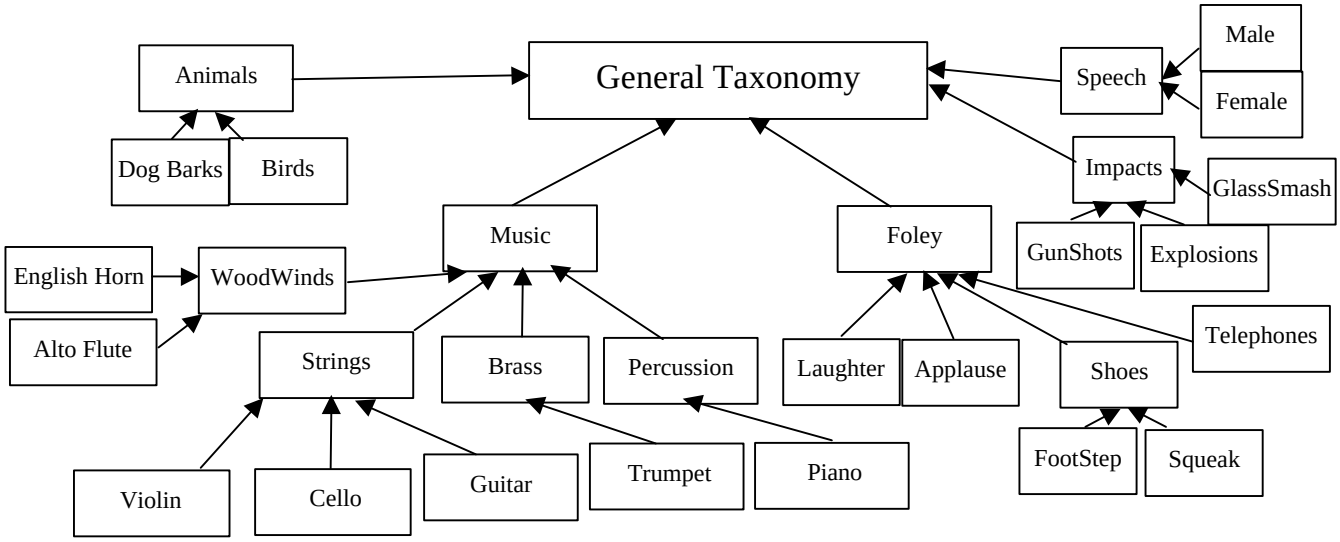

Figure 9. A hierarchical taxonomy including both musical and non-musical sources.

B. Probability Model Classifiers

A number of probability model description schemes are defined in MPEG-7. The motivation for standardization is driven by the cost of designing and training a robust classifier; which can be very computationally intensive for large-scale applications. Statistical models for classification of content may be shared via a standard interface to the internal probability models. With these standardized schemes in hand, it is possible to share pre-trained probability models between applications even if the specific extraction methods vary, thus allowing widespread re-use of models. The following sections outline the use of probability model description schemes for sound recognition.

1) Finite State Models

Sound phenomena are dynamic. The spectral features vary in time and it is this variation that gives a sound its characteristic fingerprint for recognition. MPEG-7 sound-recognition models partition a sound class into a finite number of states based on the spectral features; individual sounds are described by their trajectories through this state space. Each state is modeled by a continuous probability distribution such as a Gaussian.

The dynamic behaviour of a sound class through the state space is modeled by a k x k transition matrix that describes the probability of transition to each of the k states in a model given a current state. For a transition matrix, **T**, the ith row and jth column entry is the probability of transitioning to state j at time t given state i at time $t-1$.

An initial state distribution, which is a k x 1 vector of probabilities, is also required for a finite-state model. The kth element in the vector is the probability of being in state k in the first observation frame.

2) Multi-dimensional Gaussian Distributions

The multi-dimensional Gaussian distribution is used for modeling the states. Gaussian distributions are parameterized by a *1 x n* vector of means, **m**, and an *n x n* covariance matrix, **K**, where *n* is the number of features (columns) in the sound observation vectors. The expression for computation of probabilities for a random column vector, **x**, given the Gaussian parameters is:

$$f_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2p)^{\frac{n}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \mathbf{K}^{-1}(\mathbf{x}-\mathbf{m})\right].$$

3) Continuous Hidden Markov Models

A continuous hidden Markov model is a finite state model with Gaussian distributions approximating each state's probability distribution. The states are *hidden* since we are not given the states along with the data. Rather, we must use the observable data to infer the hidden states. The states are clusters in the feature space of the sound data; namely, the `SpectrumBasisProjection` audio descriptor discussed earlier. Each row of the projected feature matrix, defined above, is a point in an n-dimensional vector space. The cloud of points is divided into multiple states (Gaussian clusters) using maximum *a posteriori* (MAP) estimation. The MAP estimator has the property of minimizing the entropy, or degree of uncertainty, of the model whilst maximizing the number of bits of evidence (information) that supports each model parameter, (Brand,1998; Brand 1999) Figure 10 shows a representation of 4 Gaussian-distributed states (vector point clouds) in two dimensions.

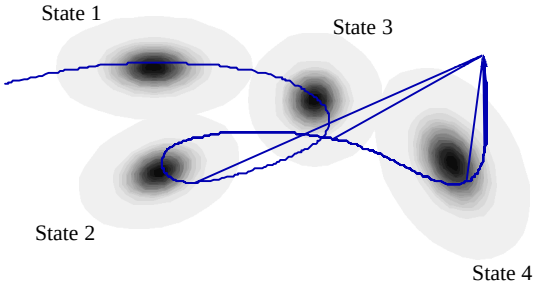

Figure 10. Four estimated Gaussian states depicted in a two-dimensional vector space. Darker regions have higher probabilities. Sounds are represented as trajectories in such a vector space, the states are chosen to maximize the probability of the model given the observable evidence; i.e. the training data. The line shows a possible trajectory of a sound vector through the space.

4) MPEG-7 representation of hidden Markov models

DDL example 2 illustrates the use of probability model description schemes for representing a continuous hidden Markov model with Gaussian states; in this example floating-point numbers have been rounded to 2 decimal places for display purposes only.

```
<ProbabilityModel xsi:type="ContinuousMarkovModelType" numberStates="7">
<Initial dim="7">
0.04  0.34  0.12  0.04  0.34  0.12  0.00 </Initial>
<Transitions dim="7 7">
0.91  0.02  0.00  0.00  0.05  0.01  0.01
0.01  0.99  0.00  0.00  0.00  0.00  0.00
0.01  0.00  0.92  0.01  0.01  0.06  0.00
0.00  0.00  0.00  0.99  0.01  0.00  0.00
0.02  0.00  0.00  0.00  0.97  0.00  0.00
0.00  0.00  0.01  0.00  0.00  0.98  0.01
0.02  0.00  0.00  0.00  0.00  0.02  0.96
</Transitions>
<State><Label>1</Label></State>
<!—State 1 Observation Distribution -- >
<ObservationDistribution xsi:type="GaussianDistributionType">
<Mean dim="6">
5.11 -9.28 -0.69 -0.79  0.38  0.47
</Mean>
<Covariance dim="6 6">
1.40 -0.12 -1.53 -0.72  0.09 -1.26
-0.12  0.19  0.02 -0.21  0.23  0.17
```

```
-1.53  0.02  2.44  1.41 -0.30  1.69
-0.72 -0.21  1.41  2.27 -0.15  1.05
0.09  0.23 -0.30 -0.15  0.80  0.29
-1.26  0.17  1.69  1.05  0.29  2.24
</Covariance>
<State><Label>2</Label></State>
<!—Remaining states use same structures-- >
<\PobabilityModel>
```

DDL Example 2. Instantiation of a Probability Model in the MPEG-7 DDL language. The model parameters were extracted using a maximum *a posteriori* estimator. The description scheme represents the initial state distribution, transition matrix, state labels, and individual Gaussian means and covariance matrices for the states.

IV. Sound Classification, Similarity and Example Search Applications

A. Classification Application

We trained 19 HMMs, using MAP estimation, on a large database (1000+ sounds) divided into 19 sound classes as described by the leaf nodes in the general sound taxonomy shown in Figure 9 above. The database was split into separate training and testing data sets. That is, 70% of the sounds were used for training the HMM models and 30% were used to test the recognition performance of the models on novel data. Each sound in the test set was presented to all 19 models in parallel, the HMM with the maximum likelihood score, using a method called Viterbi decoding, was selected as the representative class for the test sound; see Figure 11.

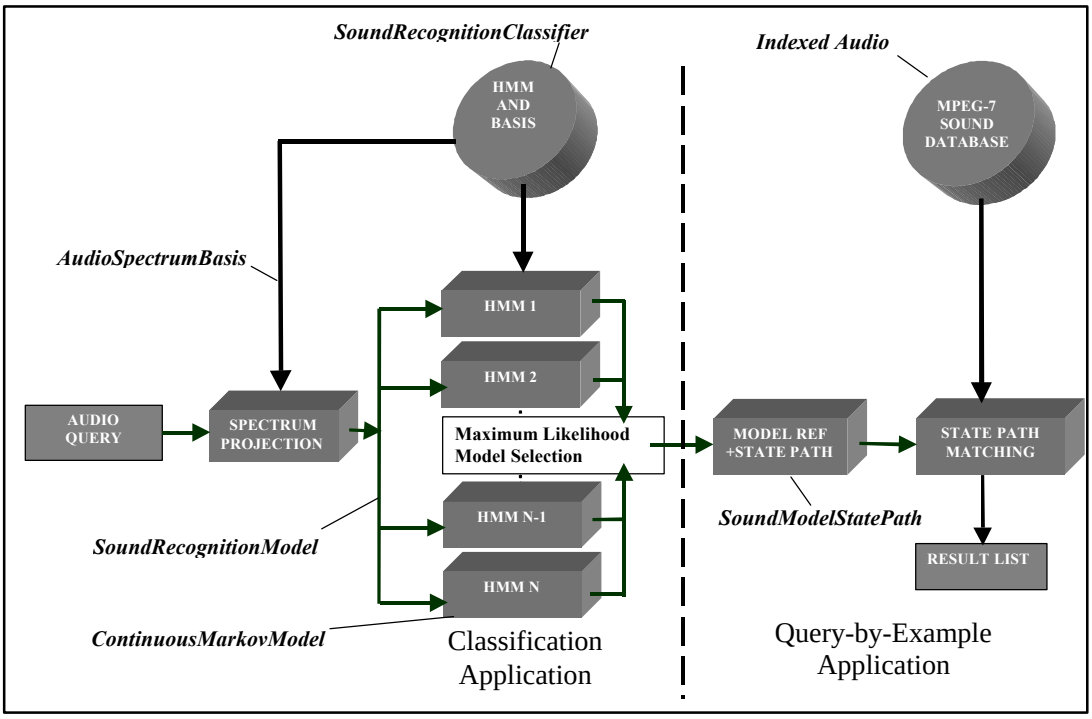

Figure 11: Sound Classification and Similarity System Using Parallel HMMs

The results of classification performance on testing data are shown in Table 1. The results indicate very good recognizer performance across a broad range of sound classes. Of particular note is the ability of the classifiers to discriminate between speech sounds and non-speech sounds including distinguishing between male and female speakers. The between class discrimination indicates a high degree of category resolution for the system.

Table 1. Performance of 19 classifiers trained on 70% and cross-validated on 30% of a large sound database. The mean recognition rate indicates high recognizer performance across all the models..

Model Name	% Correct Classification
[1] AltoFlute	100.00
[2] Birds	80.00
[3] Pianos (Bosendorfer)	100.00
[4] Cellos (Pizz and Bowed)	100.00
[5] Applause	83.30
[6] Dog Barks	100.00
[7] English Horn	100.00
[8] Explosions	100.00
[9] Footsteps	90.90
[10] Glass Smashes	92.30
[11] Guitars	100.00
[12] Gun shots	92.30
[13] Shoes (squeaks)	100.00
[14] Laughter	94.40
[15] Telephones	66.70
[16] Trumpets	80.00
[17] Violins	83.30
[18] Male Speech	100.00
[19] Female Speech	97.00
Mean Recognition Rate	**92.646**

B. Generalized Sound Similarity

In addition to classification, it is often useful to obtain a measure of how *close* two given sounds are in some perceptual sense. It is possible to leverage the internal, hidden, variables generated by an HMM in order to compare the evolution of two sounds through the model's state space. For each input query sound to a HMM, the output is a series of states through which sound passed. Each sampled state is given a *likelihood* that is used to cumulatively compute the probability that the sound actually belongs to the given model. The `SoundModelStatePath` descriptor contains the dynamic state path of a sound through a HMM model. Sounds are indexed by segmentation into model states or by sampling of the state path at regular intervals. Figure 12 shows a spectrogram of a dog bark sound with the state path through the "DogBark" HMM shown below.

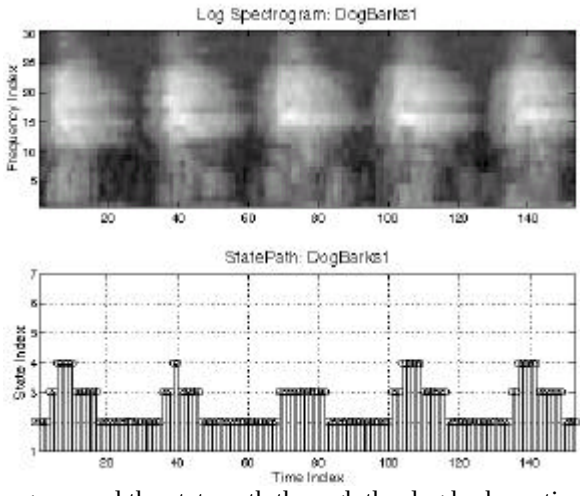

Figure 12. Dog bark spectrogram and the state path through the dog bark continuous hidden Markov model

The state path is an important method of description since it describes the evolution of a sound with respect to physical states. The state path shown in the figure indicates physical states for the dog bark; there are clearly delimited onset, sustain and termination/silent states. This is true of most sound classes; the individual states within the class can be inspected via the state path representation and a useful semantic interpretation can often be inferred.

There are many possible methods for computing similarity between state paths; dynamic time warping and state histogram sum-of-square errors are two such methods. Dynamic time warping (DTW) uses linear programming to give a distance between two functions in terms of the cost of warping one onto the other. We may apply DTW to the state paths of two sounds in order to estimate the similarity of their temporal evolutions. However, there are many cases where the temporal evolution is not as important as the relative balance of occupied states between sounds. This is true, for example, with sound textures such as rain, clapping or crowd babble. For these cases it is preferable to use a temporally agnostic similarity metric such as the sum-of-square errors between state occupancy histograms. These similarity methods may be applied to a wide variety of sound classes and thus constitute a generalized sound similarity framework.

C. Query-by-Example Application

The system shown in the right-hand side of Figure 11 implements a query-by-example application. The audio feature extraction process is applied to a target query sound, namely spectrogram projection against a stored set of basis functions for each model in the classifier. The resulting dimension-reduced features are passed to a Viterbi decoder for the given classifier and the HMM with the maximum-likelihood score for the given features is selected. The model reference and state path are recorded and the results are matched by comparing the state path to the state paths of all the sounds for the given class in a pre-computed MPEG-7 index database.

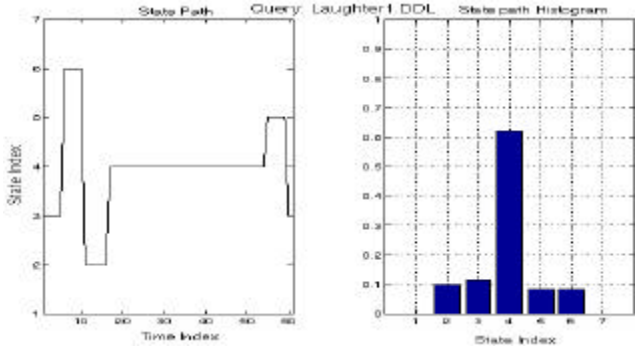

Figure 13. query sound represented by a state-path histogram for the Laughter HMM.

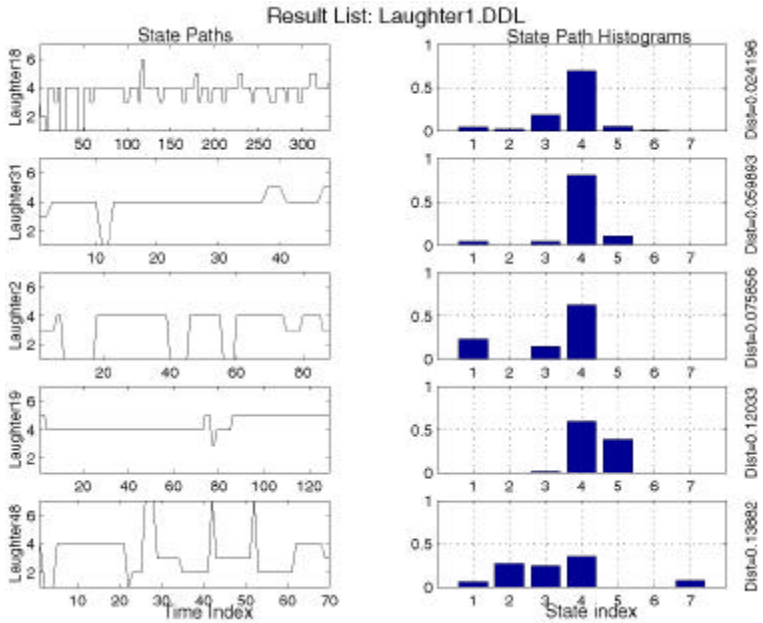

Figure 14. 5-best matches for the query sound. The distances between the target sound and the result sounds are given on the right-hand side of the figure. These distances were computed using the sum of square errors between the state-path histograms.

Figure 13 shows a query sound (Laughter) and Figure 14 shows the resulting closest matches using the difference in state-path occupancy histograms. The state paths and the histograms are also shown in the figures as well as the resulting distance estimates for each of the returned matches.

D. *Non-Categorical Similarity Ratings*

Using such similarity measures it is possible to automatically organize sonic materials for a composition. The examples given above organize similarity rankings according to a taxonomy of categories. However, if a non-categorical interpretation of similarity is required one may simply train a single HMM, with many states, using a wide variety of sounds. Similarity may then proceed without category constraints by comparing state-path histograms in the large generalized HMM state space.

V. CONCLUSIONS

In this paper we have outlined some of the tools that are available within the MPEG-7 standard for managing complex sound content. In the first part of the paper we presented independent subspace analysis as a method for performing analysis and re-synthesis of individual sources in a mixed audio file. We also showed that ISA may be used to obtain statistically salient features that may be applied with great generality to sound recognition and sound similarity tasks.

One of the major design criteria for the tools was the ability to analyze and represent a wide range of acoustic sources including textures and mixtures of sound. The tools presented herein exhibited good performance on musical sounds as well as traditionally non-musical sources such as vocal utterances, animal sounds, environmental sounds and sound effects. Amongst the applications presented were robust sound recognition using trained probability model classifiers and sound similarity matching using internal probability model state variables.

In conclusion, the description schemes and extractor methodologies outlined in this paper provide a consistent framework for analyzing, indexing and querying sounds from a wide range of different classes. These tools have been made widely available as a component of the reference software implementation of the MPEG-7 standard. It is hoped that the ability to manipulate sound in novel ways and the ability to search for “sounds like” candidates in a large database of sounds will become important new tools for sound-designers, composers and many other users of new music technology.

References

Bell, A. J. and Sejnowski, T.J. 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation,* 7:1129-1159.

Boreczky, J.S. and Wilcox, L.D. 1998. A hidden Markov model framework for video segmentation using audio and image features, in *Proceedings of ICASSP’98,* pp.3741-3744, Seattle, WA.

Brand, M. 1998. Structure discovery in conditional probability models via an entropic prior and parameter extinction. *Neural Computation.*

Brand, M. 1999. Pattern discovery via entropy minimization. In *Proceedings, Uncertainty’99.* Society of Artificial intelligence and Statistics #7. Morgan Kaufmann.

Cardoso, J.F. and Laheld, B.H. 1996. Equivariant adaptive source separation. *IEEE Trans. On Signal Processing,* 4:112-114.

Casey, M.A., and Westner, A. 2000. Separation of mixed audio sources by independent subspace analysis. *Proceedings of the International Computer Music Conference,* ICMA, Berlin.

Hyvarinen, A. 1999. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. On Neural Networks,* 10(3):626-634.

Martin, K. D. and Kim, Y. E. 1998. Musical instrument identification: a pattern-recognition approach. Presented at the 136th Meeting of the Acoustical Society of America, Norfolk, VA.

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

Zhang, T. and Kuo, C. 1998. Content-based classification and retrieval of audio. SPIE 43rd Annual Meeting, *Conference on Advanced Signal Processing Algorithms, Architectures and Implementations VIII,* San Diego, CA.