Return to Index

Abstract— the MPEG-7 international standard contains new tools for computing similarity and classification of audio clips and for

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-

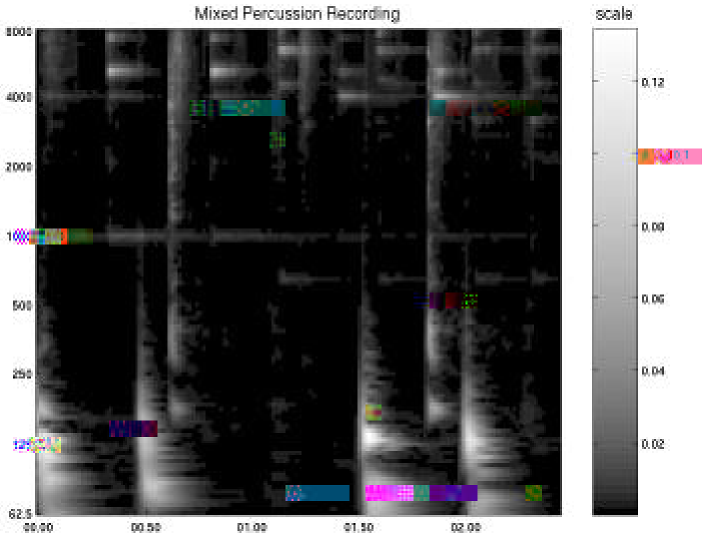


Figure 2. Log-frequency power spectrum of a mixed percussion recording.

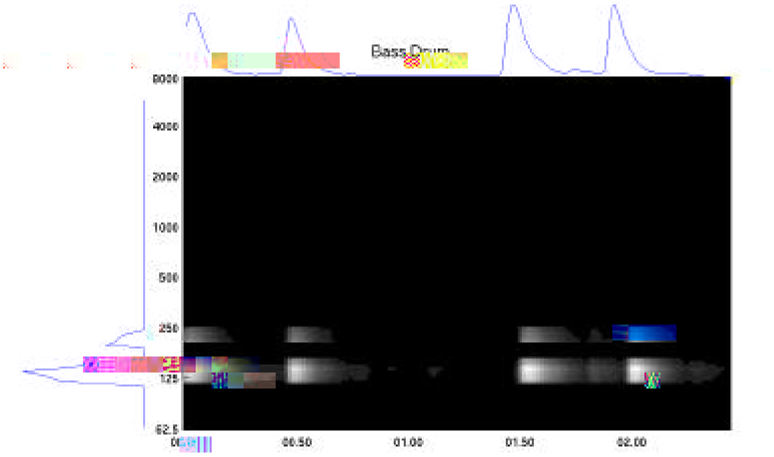


Figure 3. Spectrogram reconstruction of the bass drum estimated by re-filtering the input spectrogram using ISA basis functions.The function to the left is frequency mask component and the function across the top is the time masking component.

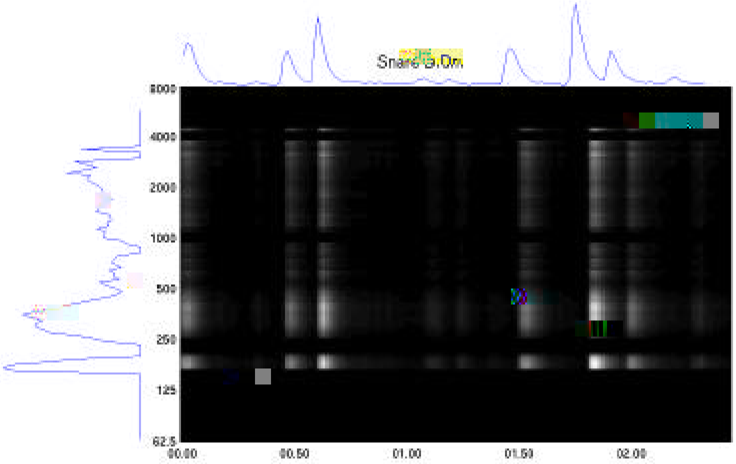


Figure 4. Masking functions and spectrogram reconstruction of the snare drum.

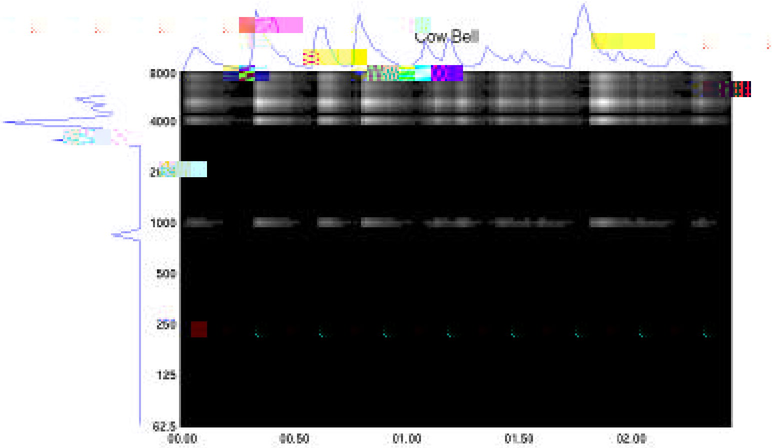


Figure 5. Masking functions and spectrogram reconstruction of the cow bell.

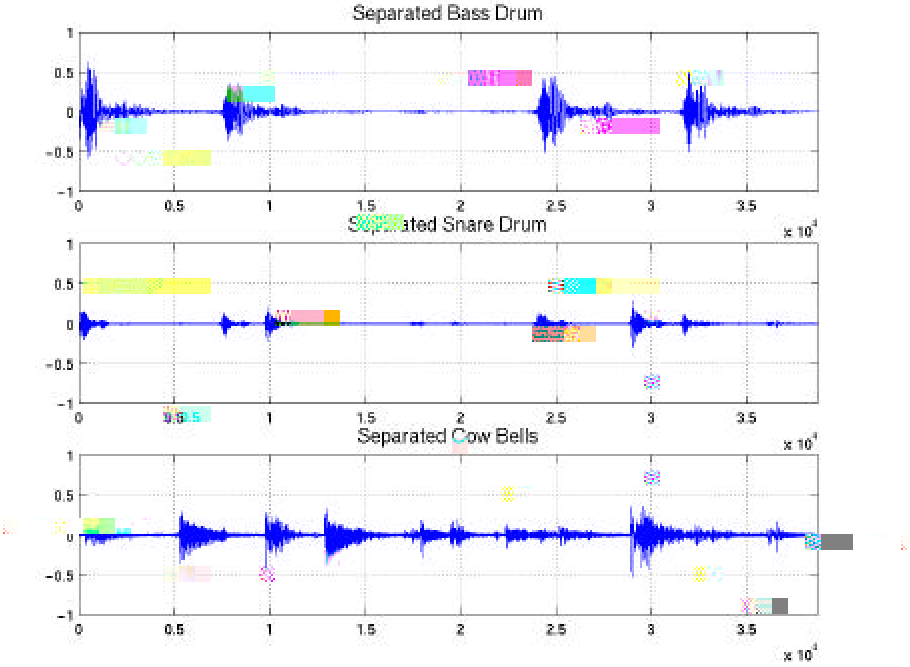


Figure 6. Separated audio signals using ISA basis functions with spectrogram re-filtering.

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

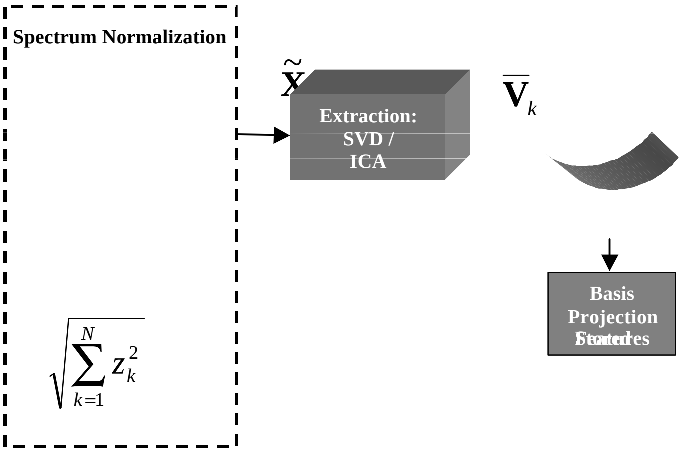
$$I(K) = \frac{\sum_{i=1}^{K} S(i,i)}{\sum_{j=1}^{N} S(j,j)}$$

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.



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.

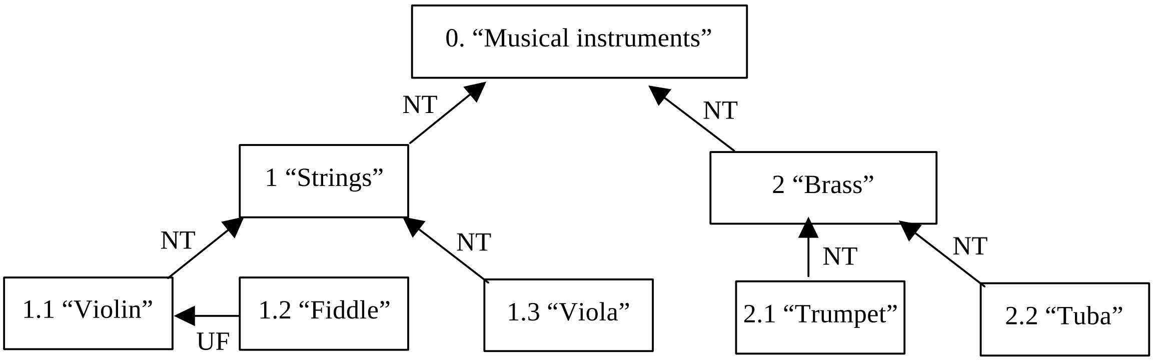


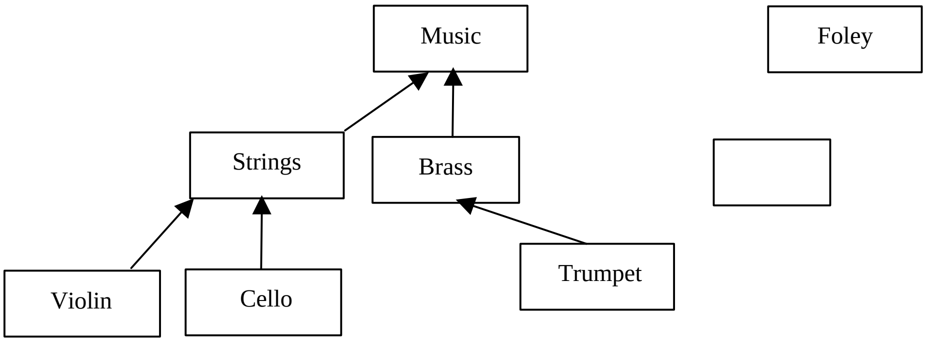
Figure 8. A controlled-term taxonomy of part of the *Musical Instruments* hierarchy

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.



Strings

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:

vectors.48he exp7nuou 10.08 toj -fn t Tj sn

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

3) *Continuous Hidden Markov Models*

A.2333323oushdels*Continu53 Hidden M67 /F..2s, given 57 matrix,445kov Modsi /F0we 498 notrs is:*

```
-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.

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.

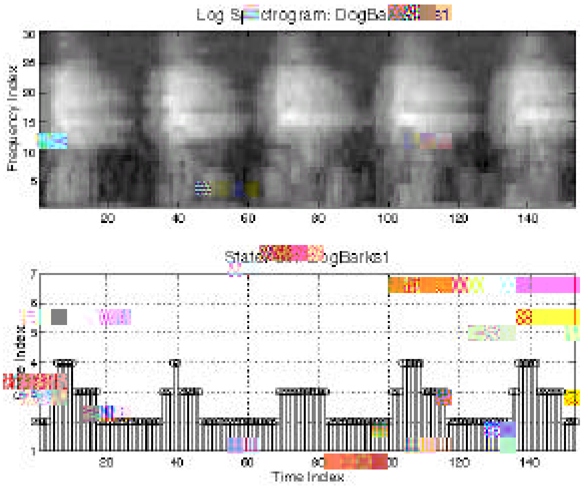


Figure 12. Dog bark spectrogram and the state path through the dog bark continuous hidden Markov model

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-

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.