Similarity Search in Animal Sound Databases

Rolf Bardeli

***Abstract*—In the past, similarity search for audio data has largely been focused on music. Recent digitization efforts in some of the larger animal sound archives bring other types of audio recordings into the focus of interest. Although recordings in animal sound archives are usually very well annotated by metadata, it is almost impossible to manually annotate all sounds made by animals in each recording. Complementary to classical text-based querying of databases that exploit available annotations, algorithms capable of automatically finding sections of recordings similar to a given query fragment provide a promising approach for *content-based* navigation. In our work, we present algorithms for feature extraction, as well as indexing and retrieval of animal sound recordings. Making use of a concept from image processing, the structure tensor, our feature extraction algorithm is adapted to the typical curve-like spectral features that are characteristic for many types of animal sounds. We propose a method for similarity search in animal sound databases which is obtained by adding a novel ranking scheme to an existing inverted file based approach for multimedia retrieval. Evaluation of our methods is based on recordings from the Animal Sound Archive, Berlin.**

***Index Terms*—Similarity search, animal sounds, feature extraction, indexing, retrieval, structure tensor.**

I. Introduction

DURING the last years, animal sound archives (such as the Macaulay Library of Natural Sounds, the British Library Sound Archive, and the Animal Sound Archive in Berlin) have started to put large efforts into the digitization of their analogue audio material [1]. These efforts, which have been mainly started for conservation reasons, also simplify access to recordings and make them accessible for algorithmic processing and analysis. Only recently, some archives have begun to create facilities for the online access of digitized audio.

For the Animal Sound Archive at Humboldt University Berlin, we have recently created a comfortable web-based user interface for access to its sound recordings. Our system furthermore allows to connect and manage external collections of animal sounds and provides means for data exchange, thus forming a platform for cooperative research in bioacoustics [2].

Finding recordings in animal sound archives is usually performed by searching metadata describing the target species of a recording, the recording location, date, and more. This metadata is often dependent on the recordists notes and expertise. For example, a recordist can only identify those vocalizations of animal species in his recordings that he is familiar with. Usually, only the target species and the loudest vocalizations in the background are annotated. Completely annotating the vast amounts of recordings in animal sound archives is a herculean task which requires good knowledge of animal voices. For all of these reasons, there is an enormous demand for methods that automatically preprocess and analyze such audio collections and extract as much higher level information as possible. This way, it would be possible to find animal vocalizations not annotated by metadata or to identify those recordings containing sounds similar to a specific example.

In this paper, we present algorithms for feature extraction, indexing and retrieval of animal sounds which meet these demands. Many animal sounds contain frequency modulated elements, resulting in curve-like time-frequency trajectories (see Figs. 1 and 6 for examples). We propose novel features tailored to the representation of such sounds. Moreover, we add a novel ranking scheme to an existing inverted file based approach for multimedia retrieval in order to allow for content-based retrieval of animal sounds. Altogether, these algorithms allow to find the exact positions in all of the indexed recordings which match a given query fragment. The results are ranked by a similarity measure induced by the features.

A. Related Work

Most of the work concerned with the algorithmic analysis of animal sound recordings focuses on classifier systems. Although less work has been carried out in the field of animal sounds than in other fields, a wide variety of methods and animal species have been examined. Already in early studies, bird song recognition with hidden Markov models (HMMs) has been proved to be a useful tool in the recognition of bird song elements [3]. In addition to birds [4], neural networks have been applied to the recognition of animal species like killer whales [5] and crickets [6]. Self-organising maps have also been used for various classification tasks concerning animal sounds [7], [8]. Other classifier systems have been used for bird song recognition [9]. Because of the strong tonal characteristics of many bird songs, sinusoidal modelling is a promising feature extraction step for bird song recognition and has been studied extensively in this context [10]. These approaches are usually applicable to species identification in continuous recordings as well as clustering tasks in smaller collections of recordings. Navigation in large collections of animal sounds, however, is a problem more naturally approached by similarity search methods as they have been developed in the field of multimedia retrieval.

Similarity search in general audio collections has been a research topic for several years [11]. Depending on the size of the sound database, dynamic time warping using feature similarity

Manuscript received January 14, 2008; revised September 17, 2008. Current version published January 08, 2009. This work was supported by the German Research Foundation (DFG) under Grant LIS 1-554 95 (1) Bonn uv and by the German Federal Agency for Nature Conservation (BfN) under Grant 806 82 060-K2. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xian-Sheng Hua.

The author was with the University of Bonn, 53117 Bonn, Germany, He is now with Fraunhofer IAIS, 53754 Sankt Augustin, Germany (e-mail: rolf.bardeli@gmail.com).

Digital Object Identifier 10.1109/TMM.2008.2008920

1520-9210/$25.00 © 2009 IEEE

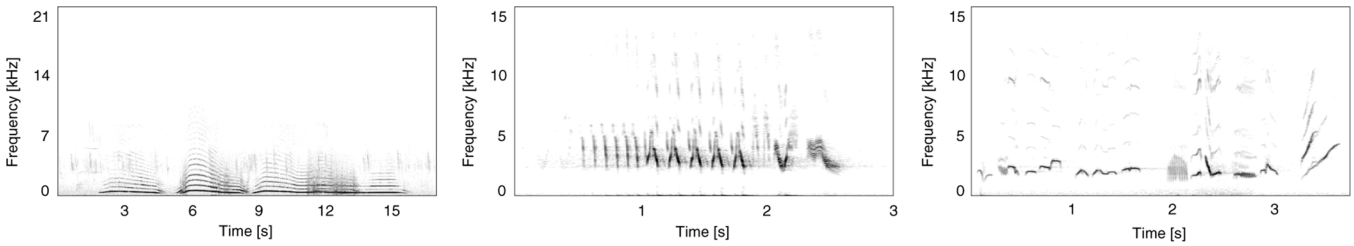

Fig. 1. Three spectrograms illustrating the typical curve-like features of animal sounds. The howling of Canis lupus (Wolf, left), the song of Fringilla coelebs (Chaffinch, middle) and a song of Turdus merula (Blackbird, right).

matrices can be used for retrieval purposes [12], [13]. Most similarity search methods, however, use sound classification as a central step. Thus, query sounds need only be compared to indexed sounds of the most likely sound classes. In this context, support vector machines [14] and hidden Markov models [15], [16] have been used as classifiers.

In some of the previous work in this context, either animal sounds have been used for evaluation or standard methods from audio similarity search have been applied to animal sound retrieval [15]–[17]. All of this work is classification based, using only general categories such as *birds*. For example, [15] uses a class for birds represented by one hidden Markov model with no more than ten states. Also, the data used in all of these cases usually comprises a small number of random recordings found in, for example, sound effect libraries or on the Internet. Animal sound archives, however, usually contain recordings from in the magnitude 1000 species of birds. Such recordings show an extreme amount of intra- and interspecific variability and it seems highly unlikely to find such kind of a simplistic representation for such a complex sound class.

Much of the work in audio similarity is focused on collections of special types of recordings, such as music. Similarity measures and search algorithms for music usually exploit its special structure. For example, the division of the octave into 12 semitones in western music is used in the definition of chroma features [18], [19]. A task that is widely regarded as solved is the identification of—possibly distorted—fragments of recordings [20]–[22]. Similarity search in music has many specialized subproblems such as audio matching [13], cover song identification [23] and genre classification [24]. Moreover, there is a special emphasis on similarity measures matching the human preception of similarity as closely as possible [25].

B. Overview

Our approach to similarity search in animal sound databases is composed of two central parts. The first one is a feature extraction algorithm leading to a compact fingerprint-like description of animal sounds. The second part comprises indexing of these features and a retrieval algorithm.

We will start our discussion of similarity search by describing the features to be extracted from each audio file which will form the search index. For each file, we first identify points of interest in the spectrum which are suitable for feature extraction. In this, our algorithm focuses its attention to positions exhibiting a strong dominant orientation of the energy distribution. These are found using a tool from image analysis: the structure tensor [26]. In a second step, we derive a feature class described by a 12-bit value from a neighborhood of each position selected in the first step. More details on feature classes are given in Section II-B.

These features are well suited for an inverted list index. In Section III, we develop a ranking technique for similarity search in animal sound databases which is obtained by adding a novel ranking scheme to an existing inverted file based approach to multimedia retrieval [20].

In Section IV, we present evaluation results of our methods on a collection of animal sound recordings from the Animal Sound Archive Berlin. For animal sounds containing curve-like spectral features, we show that species identification is often feasible. A comparison with an audio similarity search algorithm based on MPEG-7 spectrum projection features and hidden Markov models [15] shows that our method is better adapted to this kind of animal sounds. Moreover, in contrast to classification based algorithms, our method yields the exact position of a match within the audio data in addition to a sound class such as the animal species.

II. Feature Extraction

The core of most similarity search algorithms is the feature extraction process. For our similarity search algorithm, we use a two step process for feature extraction. For each file, we first identify points of interest in the spectrum which are suitable for feature extraction. In this, our algorithm focuses its attention to positions exhibiting a strong dominant orientation of the energy distribution. In a second step, we derive a feature class described by a 12-bit value from a neighborhood of each position selected in the first step.

A. Selecting Points of Interest

Most audio signals produced by animals (especially by birds) exhibit curve-like spectral components (Figs. 1 and 6). We adopt a technique from image processing, the structure tensor, in order to identify portions of the audio spectrum that show this kind of structure which is reflected by the fact that curve-like components give the spectrum a pronounced local orientation.

Interpreting grey scale images as functions $I : \mathbb{R}^2 \rightarrow \mathbb{R}, (x, y) \mapsto I(x, y)$ we can define the partial derivatives $I_x(x, y) := (\partial I)/(\partial x)(x, y)$ and $I_y(x, y) := (\partial I)/(\partial y)(x, y)$. Then, the structure tensor at position (x, y) is defined as

$$J_0(I)(x, y) := \begin{pmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{pmatrix}.$$

The structure tensor contains information about the dominant orientation of the image at a given pixel (x, y). This orientation is given by the eigenvector of $J_0(I)(x, y)$ belonging to the largest eigenvalue. Moreover, the magnitude of this eigenvalue yields some information on how pronounced this orientation is. This information, however, is easily distorted by noise because it stems only from the direct neighborhood of one pixel. In order to get more robust information, we use the mean of the structure tensors from a 11×11 pixel neighborhood of a pixel.

Now, we interpret the spectrum of an audio signal as an image and find those points where a strong dominant orientation is present. For this purpose, we compute the windowed Fourier transform of each input signal, using a Hann window of 32 milliseconds length with 50% overlap. Points of interest are computed using the logarithm of the absolute values of the transform. We then divide this spectrum into 50 equally spaced frequency bands. Only the neighborhoods of pixels at the center of each band are considered as possible points of interest. The width of the frequency bands is chosen such that the frequency width of the pixel neighborhoods exactly covers the band and amounts to about 300 Hz.

We will decide whether a dominant orientation is present by calculating the largest eigenvalue of the smoothed structure tensor. The mean of a set of structure tensors is a symmetric matrix

$$M := \begin{pmatrix} A & B \\ B & C \end{pmatrix} \in \mathbb{R}^{2\times2}$$

and its eigenvalues are the zeros of its characteristic polynomial $\chi_M(\lambda) = \lambda^2 - (A + C)\lambda + AC - B^2$. We now assign an *attention value* $a(x, y)$ to a given pixel (x, y) if M has two distinct eigenvalues, i.e., if the discriminant of $\chi_M, D := ((A + C)^2/4) - AC + B^2$, is larger than zero. Then, let $a(x, y) := (A + C)/(2) + \sqrt{D}$, otherwise let $a(x, y) := 0$.

The final decision whether a feature is extracted at a point (x, y) is based on a local threshold θ_1, i.e., a feature is extracted at each point with $a(x, y) > \theta_1$. In order to cope with audio signals that are too large to be kept in memory, the analysis of the audio signals is performed in a windowed manner, analysing segments of 1000 spectrogram columns and a small overlap corresponding to the width of analysis windows used for the 2-D Fourier transform. A new threshold is computed for each such segment. The threshold is found by using a histogram h of a counting the number of points in a window with values of a in a given range. More precisely, we divide the interval $[0, \max_{x,y} a(x, y)]$ into 1000 bins and define the histogram h as

$$h(\ell) := \sharp\{(x, y) \,|\, \ell \leq 1000 \cdot a(x, y)/\max_{x,y} a(x, y) \leq \ell + 1\},$$

$\ell \in \{0, 1, \ldots, 999\}$. Now, we choose ℓ such that $\sum_{i=1}^{\ell} h(1000 - i) \geq p \sum_{i=0}^{999} h(i)$. From this, we obtain θ_1 by scaling ℓ to the range of $a : \theta_1 = (\ell/1000) \cdot \max_{x,y} a(x, y)$. A practical value is $p = 0.96$.

The number of points of interest selected for an audio document depends on its length and complexity. Usually, 200 to 300 points of interest per second are selected. How well the points of interest correspond between signals of the same class depends on how similar the signals are. Spectrograms perceived as very similar by humans lead to a 70–90% overlap of their points of interest.

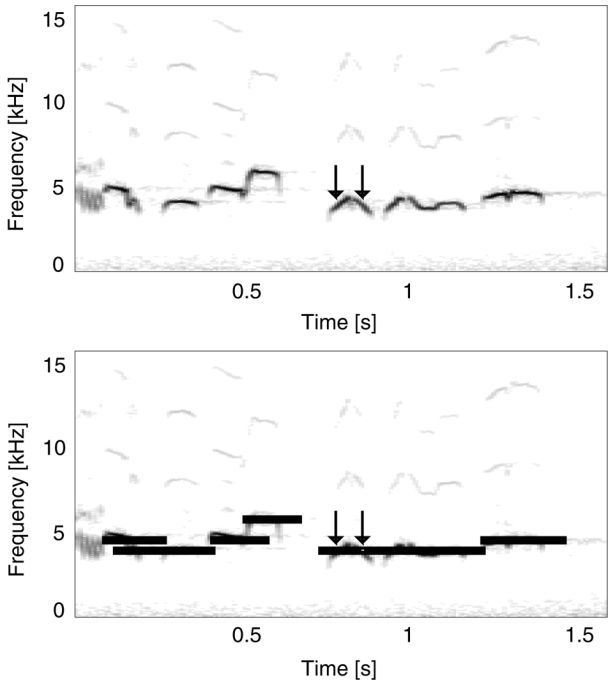

Fig. 2. Points of interest selected from an excerpt of the spectrum of Turdus merula shown in Fig. 1. Arrows mark positions that are used for illustrating the computation of feature classes in Fig. 3.

Fig. 2 shows points of interest selected from the spectrum of a Blackbird song.

B. Feature Classes

We now proceed to assigning a feature $(i, f, t, c) \in \mathbb{Z}^4$ to each position (f, t) in the spectrum of the signal D_i determined by the procedure described in the previous section. The frequency-band f and the time-position t are determined by the position in the spectrum. In the following, we will describe how the feature class c is derived.

The feature class c is a 12-bit value, each bit describing an aspect of the geometric shape of the 2-D Fourier transform of a 16×16 pixel neighborhood of a point of interest. Here, the size of the neighborhood has been chosen as a power of 2 for computational reasons. The feature class of a point p is obtained as follows (see Fig. 3). First, compute the discrete 2-D Fourier transform of the 16×16 pixel neighborhood of the point, containing the point p at the upper left pixel of its central four pixels [Fig. 3(a) and (d)]. This neighborhood is divided into four columns, each 4 pixels wide [Fig. 3(b) and (e)]. The four subcolumns of each column are concatenated to four vectors $v_1, \ldots, v_4$ of length 64. The six possible pairwise scalar products $v_i^\top v_j (i \neq j)$ of the v_i comprise entries one to six of a new vector $s \in \mathbb{R}^{12}$. Entries seven to twelve are obtained similarly, considering rows instead of columns. Unfortunately, the representation of images by the 2-D Fourier transform is quite unintuitive and hence it is hard to interpret the characteristics of an image from its transform.

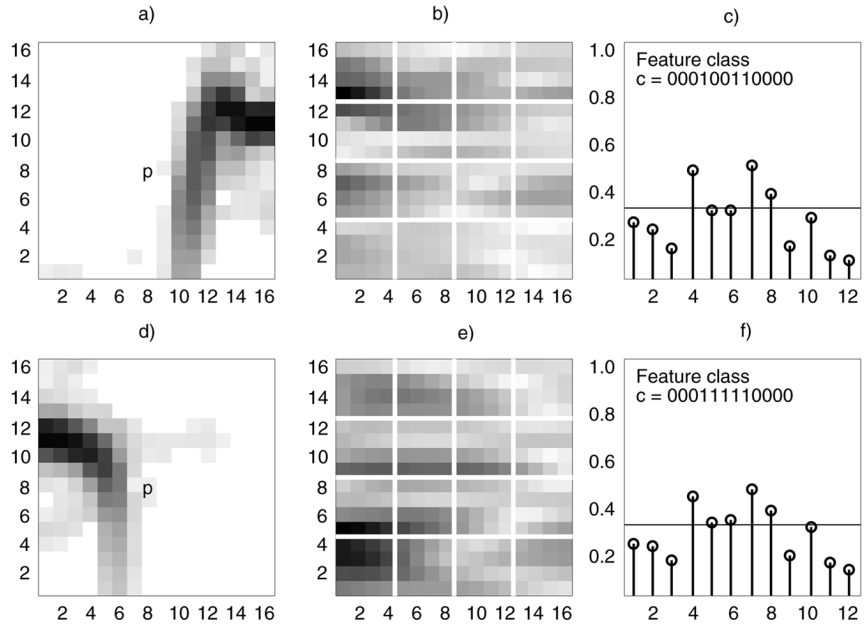

Fig. 3. Deriving the feature class from the neighborhood of a pixel p. Parts a) and d) give examples of such neighborhoods from Fig. 2. The respective 2-D Fourier transforms are given in parts b) and e). These are divided into 4×16 element rows and columns whose pairwise dot products constitute the histogram representations given in parts c) and f). The feature classes in the latter parts are found by assigning a bit 1 or 0 to each histogram value depending on whether it is larger than the threshold (horizontal line) or not.

Finally, the feature class c is obtained by first normalising s with respect to the Euclidean norm and then setting bit i of c to 1 if $s_i > 0.4$. All other bits are set to zero (Fig. 3(c) and (f)[1]).

Using the shape of the Fourier transform to define the feature class gives some robustness towards small changes of the pixel position from which the feature class is derived. Reducing the information from the Fourier transform to a bit-feature allows us to define the distance between two feature classes as the number of differing bits. This is very easy to compute and allows fast retrieval of all features similar to a given feature class from a database. Altogether, this bit-feature gives a compact, translation insensitive, and easy to compare local description of the audio spectrum.

III. Indexing and Retrieval

The features described in the previous section are well suited for the use in inverted list-based retrieval algorithms. We derive our similarity search algorithm from a general framework for multimedia retrieval [27]. In our setting, a specialized version of the framework can be formulated as follows.

Let $\mathcal{D} := (D_1, \ldots, D_n)$ be a database of audio files. In the following, we will refer to the D_i as documents. After feature extraction, each document D_i is described by a set of features $F[D_i] = \{(t_1^i, f_1^i, c_1^i), \ldots, (t_{n_i}^i, f_{n_i}^i, c_{n_i}^i)\}$. As in Section II.B, these features comprise time position t_j^i, frequency band f_j^i and feature class c_j^i.

Accordingly, a query signal Q is described by a set of features $F[Q] = \{(t_1^Q, f_1^Q, c_1^Q), \ldots, (t_{n_Q}^Q, f_{n_Q}^Q, c_{n_Q}^Q)\}$. Using a very strict definition of similarity, we look for all ways in which we can translate the time components of $F[Q]$ by a time offset t such that the t-shifted version $F[Q]+t$ becomes a subset of a document feature set $F[D_i]$. More formally, we are interested in the set $G_{\mathcal{D}}(F[Q]) = \{(t,i) \,|\, F[Q]+t \subset F[D_i]\}$. Here, $F[Q]+t := \{(t^Q+t, f^Q, c^Q) \,|\, (t^Q, f^Q, c^Q) \in F[Q]\}$. This means we are looking for all pairs (t,i) such that the query features shifted by t are contained in the features of document D_i. Fig. 4 gives an illustration of this definition.

As a first step, we can answer the query Q by intersecting the results for each element in Q:

$$G_{\mathcal{D}}(F[Q]) = \bigcap_{q \in F[Q]} G_{\mathcal{D}}(\{q\}). \tag{1}$$

This is easily seen by substituting the definition of $G_{\mathcal{D}}(F[Q])$ into (1). Equation (1) by itself, however, does not give us an efficient retrieval algorithm. It would however allow us to evaluate the query if we had the retrieval results $G_{\mathcal{D}}(\{q\})$ for each q at hand. Now, we can store a finite subset of these results in the form of inverted lists and easily recover the other lists from this index. This can be seen from the simple observation that $G_{\mathcal{D}}(\{(t,f,c)\})$ can be recovered from $G_{\mathcal{D}}(\{(0,f,c)\})$ by translating all time positions by t. Thus, we can evaluate queries by evaluating the intersection:

$$G_{\mathcal{D}}(Q) = \bigcap_{(t,f,c) \in Q} G_{\mathcal{D}}(\{(0,f,c)\}) - t. \tag{2}$$

The sets $G_{\mathcal{D}}(\{(0,f,c)\}) - t$, however, are actually inverted lists containing all pairs (i,t') of time position t' and document index i such that feature class c is present in frequency band f in document D_i. So, if these inverted lists are stored in an index for all combinations (f,c) contained in the document collection, the index is all that is needed to answer queries in the strict sense defined above. In this case, the time complexity for retrieval only depends on the length of the inverted lists. More precisely,

[1]A threshold of 0.3 instead of 0.4 is used in the figure because it gives a better illustration of the thresholding process.

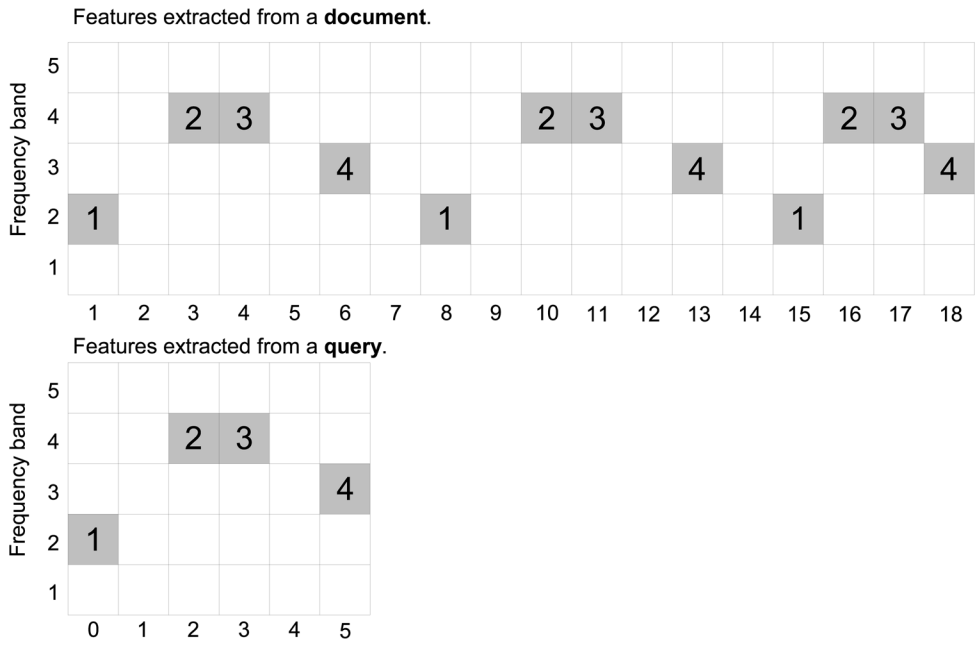

Fig. 4. Features extracted from a document D_1 and a query Q. The query matches the document at positions 1 and 8 but not at position 15. Thus in this case we have $G_{\mathcal{D}}(F[Q]) = \{(1,1),(1,8)\}$.

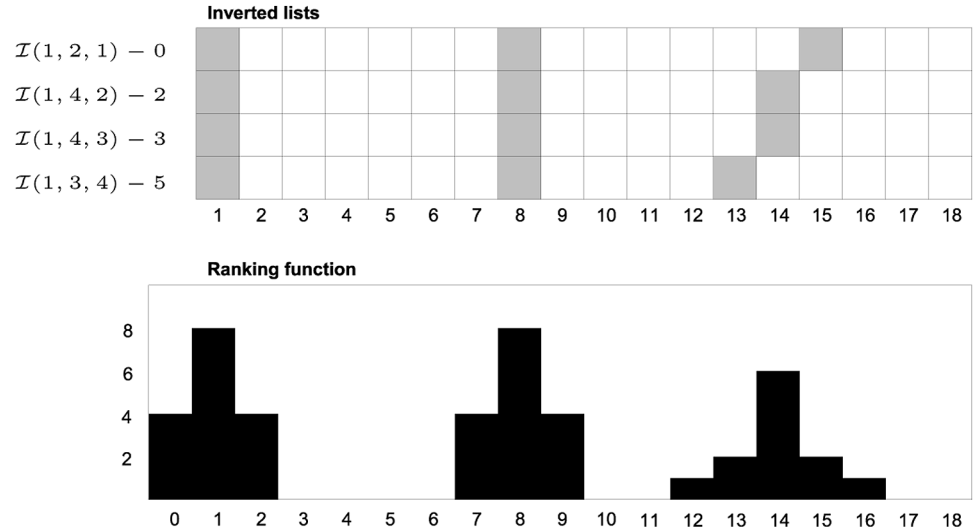

Fig. 5. Inverted list and ranking function for the document and query of Fig. 4. $\mathcal{I}(1,f,c) - t$ denotes the inverted list for frequency-band f and feature class c with t subtracted from each entry of the list. The local ranking function r for this example is $r(-1) = 1, r(0) = 2, r(1) = 1$ and $r(x) = 0$ otherwise. The inverted lists are perfectly aligned at positions 1 and 8, corresponding to the exact matchings of the query. The query features are similar to the document features near position 15, resulting in a broad peak in the ranking function.

the time complexity is $O(|F[Q]|\ell_{\min}\log\ell_{\max})$ where $\ell_{\min}$ resp. $\ell_{\max}$ is the length of the shortest resp. longest list. Fig. 5 shows the inverted lists for the example of Fig. 4 adjusted according to the query. Intersecting these lists gives the exact matches of the query.

Our approach to similarity search in animal sound databases is derived from this framework. We will however replace the strict process of intersecting inverted lists by a more flexible ranking process.

As we have seen, with feature extraction producing index objects (i,f,t,c), indexing simply amounts to storing all the index objects obtained from a document collection in a relational database table with one column for each of the index elements i, f, t, and c. At the time of query processing, this allows us to restore inverted lists $\mathcal{I}(i,f,c)$ for each triple (i,f,c) consisting of all time positions at which feature class c was detected in frequency band f of document D_i. We can speed up this process enormously by creating an index on the database columns for i, f, and c within the relational database.

Using the framework described above, in order to answer a query $F[Q] = \{(f_1,t_1,c_1),\ldots,(f_q,t_q,c_q)\}$ we would have to carry out the process described by (2). This means that we would retrieve, for each document D_i, the inverted lists for each query element $(f,t,c) \in Q$, shift the corresponding time values by t

TABLE I
RETRIEVAL RESULTS FOR FIVE SONGS OF EACH OF TEN SPECIES OF BIRDS. FOR EACH QUERY RECORDING, WE GIVE ITS LENGTH, THE HIGHEST RANK OF A RECORDING OF THE SAME SPECIES WITHIN THE RETRIEVAL RESULTS AND THE QUOTIENT OF THE RANKING VALUE OF EACH HIT WITH RESPECT TO THE HIGHEST RANKING VALUE FOR THE QUERY. THE RANK OF A HIDDEN MARKOV MODEL REPRESENTING THE QUERY SPECIES ACCORDING TO LOG-LIKELIHOOD IS GIVEN IN BRACKETS IN THE THIRD COLUMN

query number	query length	highest rank of same species	relative rank
Acrocephalus arundinaceus (Great Reed Warbler)			
1	3.7s	58 (135)	0.616
2	5.4s	70 (131)	0.61
3	7.6s	99 (17)	0.6
4	6.5s	43 (11)	0.68
5	4.7s	135 (19)	0.54
Troglodytes troglodytes (Wren)			
6	4.2s	2 (70)	0.95
7	7.1s	83 (56)	0.51
8	16.4s	4 (52)	0.75
9	6.9s	3 (43)	0.85
10	6.5s	10 (35)	0.61
Fringilla coelebs (Chaffinch)			
11	3.3s	10 (32)	0.74
12	3.6s	1 (8)	1
13	4.5s	2 (11)	0.97
14	3.3s	1 (25)	1
15	3.3s	1 (14)	1
Lullula arborea (Wood Lark)			
16	2.2s	36 (63)	0.57
17	3.8s	1 (7)	1
18	4.0s	4 (4)	0.96
19	4.0s	7 (41)	0.85
20	4.6s	2 (4)	0.98
Parus caeruleus (Blue Tit)			
21	2.5s	10 (56)	0.75
22	2.8s	15 (131)	0.76
23	5.1s	18 (5)	0.63
24	2.2s	1 (32)	1
25	3.5s	83 (54)	0.51
Turdus merula (Blackbird)			
26	3.5s	41 (16)	0.64
27	3.8s	17 (8)	0.67
28	3.7s	43 (5)	0.71
29	4.0s	11 (5)	0.87
30	4.5s	41 (17)	0.6
Phylloscopus trochilus (Willow Warbler)			
31	5.0s	1 (15)	1
32	4.4s	1 (13)	1
33	5.3s	1 (1)	1
34	7.3s	1 (7)	1
35	18.0s	1 (2)	1
Phylloscopus sibilatrix (Wood Warbler)			
36	3.9s	7 (67)	0.87
37	4.6s	1 (5)	1
38	6.3s	1 (4)	1
39	7.6s	2 (6)	0.79
40	5.8s	3 (2)	0.87
Parus major (Great Tit)			
41	11.5s	504 (6)	0.49
42	4.8s	334 (21)	0.45
43	4.2s	376 (37)	0.34
44	5.1s	104 (35)	0.66
45	3.2s	263 (30)	0.53
Ailuroedus buccoides (White-eared Catbird)			
46	5.4s	18 (9)	0.75
47	2.9s	27 (1)	0.68
48	3.8s	4 (1)	0.86
49	7.5s	11 (1)	0.7
50	5.5s	11 (1)	0.71

and find the positions where the query matches the document by intersecting these adjusted lists.

We use two mechanisms to make this scheme suitable for similarity search instead of exact matching. First, for a triple $(f,t,c) \in Q$, we merge all inverted lists $\mathcal{I}(i,f,c')$ having a feature class c' differing from c in at most Δ bits, where Δ is a parameter adjustable by the user. Second, we replace exact list intersection by a ranking scheme as follows. For each document D_i, we will compute a ranking function $r_i : \mathbb{Z} \to \mathbb{Z}$. For each time position t in document $D_i, r_i(t)$ describes how similar the query is to the document at that position.

For each time position t in one of the inverted lists fetched for a query element, we add a local ranking function $r : \mathbb{Z} \to \mathbb{Z}$ to r_i centered at position t. More precisely, if the q element query Q is defined as above, we retrieve one inverted list ℓ_{ij} for each document D_i and each query element (f_j, t_j, c_j), where $j \in \{1,2,\ldots,q\}$. Given these inverted lists $\ell_{i1},\ldots,\ell_{iq}$ for document D_i, we compute the ranking function r_i as follows:

$$r_i(x) = \sum_{j=1}^{q} \sum_{t \in \ell_{ij}} r(x-t).$$

Choosing $r(0) = 1$ and $r(x) = 0$ otherwise, would amount to counting the number $r_i(\tau)$ of lists containing an element at a given time position τ. Choosing a function r that is more widespread with a peak at $\tau = 0$, allows to account for variations in the time structure of a query. An example for computing the ranking function of a document is given in Fig. 5. In the example, we chose r such that $r(-1) = 1, r(0) = 2, r(1) = 1$ and $r(x) = 0$ in all other cases.

Finally, a small number of positions with high ranking values should be reported to the user. These positions are chosen in two steps.

First, peak picking is performed on the ranking function after a low pass filter has been applied. For each position t, we decide whether a peak is found at that position by regarding the ranking function at positions $t-k,\ldots,t-1$ on the one hand and the positions $t+1,\ldots,t+k$ on the other hand. The slope $m_\pm$ of the regression line minimizing $\sum_{i=1}^{k}(m_\pm i + b - r(t \pm i))^2$ is found for both parts and a peak is found whenever the slope m_- for the left part is positive and the slope m_+ for the right part is negative. The slope of the regression line is found by solving the respective normal equations. For our optimization problem, they are given by

$$A^\top A \begin{pmatrix} m_\pm \\ b \end{pmatrix} = A^\top \begin{pmatrix} r(t \pm 1) \\ \vdots \\ r(t \pm k) \end{pmatrix}, \quad \text{for } A = \begin{pmatrix} 1 & 1 \\ \vdots & \vdots \\ k & 1 \end{pmatrix}.$$

This leads to the following system of linear equations

$$\begin{pmatrix} \sum_{i=1}^{k} i^2 & \sum_{i=1}^{k} i \\ \sum_{i=1}^{k} i & k \end{pmatrix} \begin{pmatrix} m_\pm \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{k} ir(t \pm i) \\ \sum_{i=1}^{k} r(t \pm i) \end{pmatrix}$$

which, when solved for $m_\pm$, leads to

$$m_\pm = \frac{\sum_{i=1}^{k} ir(t \pm i) - \left(\sum_{i=1}^{k} r(t \pm i)\right)^2}{\frac{k}{6}(k+1)(2k+1) - k\left(\frac{k+1}{2}\right)^2}.$$

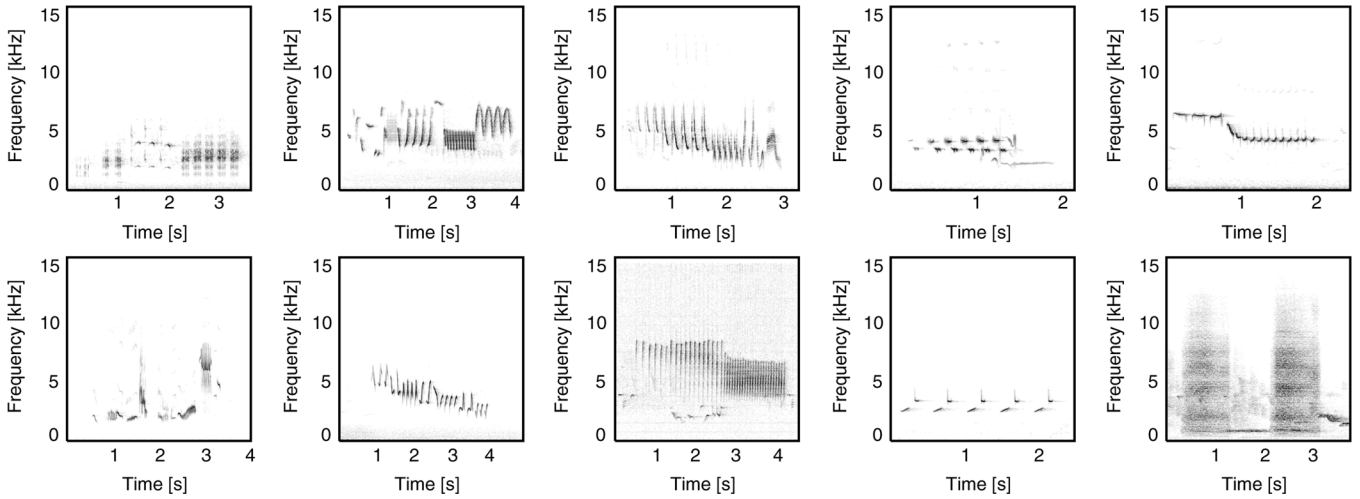

Fig. 6. Ten of the test queries showing the spectral characteristics of the different bird vocalizations used in the retrieval tests. Upper row: Acrocephalus arundinaceus (Great Reed Warbler), Troglodytes troglodytes (Wren), Fringilla coelebs (Chaffinch), Lullula arborea (Wood Lark), Parus cearuleus (Blue Tit). Lower row: Turdus merula (Blackbird), Phylloscopus trochilus (Willow Warbler), Phylloscopus sibilatrix (Wood Warbler), Parus major (Great Tit), Ailuroedus buccoides (White-eared Catbird).

After peak picking, as a second step, the ten highest ranking peaks from each document are chosen, subject to the condition that no two peaks are closer together than the query length.

IV. Results

We have tested our search algorithm on an index created from 1000 manually selected files from the Animal Sound Archive of the Humboldt University, Berlin. The indexed test dataset comprises 20 h of animal recordings from 264 species. The time required for indexing a file depends very much on the number of interest points selected. Usually, indexing takes a quarter of the playback time of a recording. The index is stored in a MySQL database and uses 260 MB of space for 1.6 GB of MPEG 1 Layer 3 compressed audio files.

It is very laborious to obtain ground truth data for data sets of the given size. A manual annotation of such data sets requires a high amount of expertise in the recognition of animal sounds as well as an extraordinary amount of time because all of the recordings have to be played back and processed in real time. We therefore tried to use for evaluation as much of the information given by the metadata as possible. Most importantly, each recording is labeled with an animal species which was the target for the recording. Additionally, many recordings come with a list of species vocalizing in the background.

For a first test, we chose ten species of birds and tested whether species identification by similarity search is possible. More specifically, we chose five songs from each bird and tested whether the correct species is found among the top ranks of the retrieval results. The test recordings where not contained in the recordings used to create the index. Fig. 6 shows the spectral characteristics of the vocalizations of the ten species used for evaluation. Table I gives a summary of the results. For each query, we have given the first position in the ranking list that belongs to a recording labeled with the species from which the query was taken. Moreover, we give the relative ranking value of this hit, i.e., the quotient of the hit's ranking value and the ranking value of the highest ranking result. The parameter $\Delta = 2$ was chosen for these tests, thus identifying two feature classes whenever they do not differ in more than 2 bits.

As a comparison, we have used a similarity search algorithm based on MPEG-7 spectrum projection features and hidden Markov models [15] that is part of the MPEG-7 standard. In order to apply this system to species identification, one hidden Markov model had to be trained for each animal species in the index set. Some of the species did not contain enough information for HMM training and were thus excluded from the comparative test. After this step, 184 of the 264 species were represented by HMMs and 900 of the 1000 audio recordings were used. In Table I we have given the rank of the HMM representing the query species in brackets in column 4. Ranking was performed by model log-likelihood.

The quality of the results of our retrieval method depends on the kind of vocalizations used as a query. By design, the feature extraction process works best for signals with curve-like spectral characteristics. This is reflected by the high ranking results for queries of vocalizations of Fringilla coelebs, Phylloscopus trochilus, Phylloscopus sibilatrix, and, to some extent, Lullula arborea and Troglodytes troglodytes. In these cases, false positives in the first ranks usually stem from structurally similar songs. For example, the songs of Phylloscopus sibilatrix and Fringilla coelebs are very similar and therefore lead to high ranking results for both species. Fig. 7 shows the ranking curve corresponding to the first 81 s of one document from the evaluation collection and the selected result positions for a query of a Troglodytes troglodytes song.

Species identification based on HMMs usually gives worse results for signals with curve-like features. It does however give excellent results for the recognition of Ailuroedus buccoides (see the lower right spectrogram in Fig. 6) whose song shows characteristics similar to human speech and is therefore well suited for the representation by spectrum projection features. In contrast to our method, no exact positions for query matches can be deduced from the HMMs.

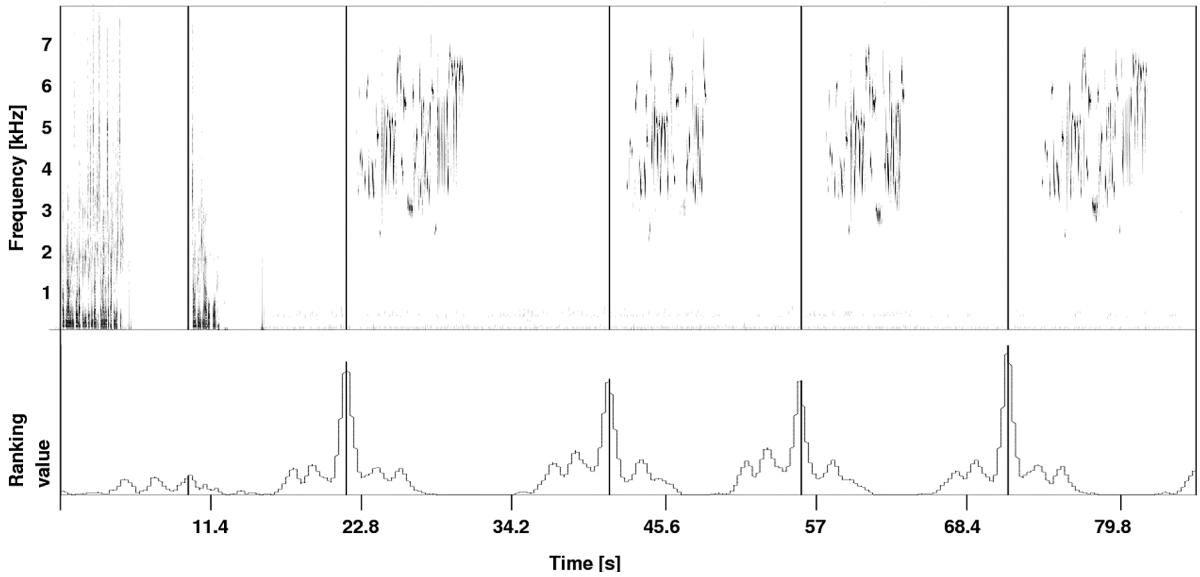

Fig. 7. Example for a retrieval result for the song of Troglodytes troglodytes. Spectrogram (top) of the first 81 seconds of a recording containing variations of this song. Vertical lines mark positions corresponding to peaks in the ranking function (bottom) selected by the algorithm.

Ailuroedus buccoides was chosen for its noise-like spectral characteristics which is not very well suited for the proposed features and is therefore well suited for finding out the limits of our method. Nevertheless, retrieval results are quite good in this case. High ranking false positives are less obviously related to queries than in the previous case. They do however tend to show either similar repetition patterns to the query or noise like structures with formant-like maxima.

Acrocephalus arundinaceus also shows noise-like features (Fig. 6, upper left spectrogram). Moreover, it has more variation in the overall structure of its songs and thus leads to worse results than Ailuroedus buccoides. It is often confused with vocalizations produced by different species such as Falco eleonores (see Fig. 8) containing repeated short noise-like elements. It also often leads to the retrieval of recordings with a strong noise background.

Turdus merula, showing spectral features that seem much better suited for our kind of features (Fig. 6, lower left spectrogram and Fig. 1, right spectrogram), leads to less favourable results. This is explained by the high variability of the blackbirds's song. As queries are required to be of a certain length in order to be discriminating enough to give reasonable results, queries for blackbird songs comprise multiple syllables of the song. Because of the high number of syllables of the blackbird and the great number of ways in which these are combined, it is improbable to find a close match of any song in the database. On the other hand it is easily confused with birds showing similar syllables and this can also be found in the results.

Particularly close investigations of the results have been carried out in order to explain the bad retrieval results for Parus major (Fig. 6, fourth spectrogram in the second row). By examining the spectra of the retrieval results, it became clear that many high ranking results usually belong to signal degradations resulting from signal clipping or starting and stopping of analogue tape recorders. These lead to sharp vertical lines in the spectrum which are detected as *curve-like features* by the feature extraction process. The vocalizations of Parus major are comprised of alternating higher and lower pitched notes. The higher pitched notes usually start with a sharp vertical line in the spectrum and thus match the signal degradations leading to similar effects.

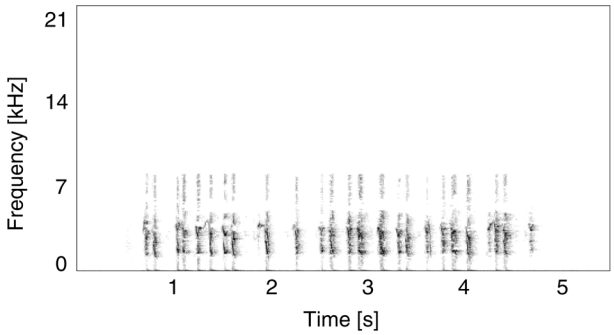

Fig. 8. Spectrogram of the song of Falco eleonorae (Eleonora's Falcon). Queries of the song of Acrocephalus arundinaceus (Fig. 6, upper left spectrogram) often lead to the retrieval of this kind of song.

V. Conclusion

Our experiments show that index-based similarity search can be a very helpful tool for navigation in animal sound databases as well as for the automatic extraction of metadata. In this paper, we have proposed algorithms for feature extraction, indexing and retrieval in animal sound databases. In particular, with our method, we can identify the exact time positions of similar portions of the indexed audio recordings.

Our features are adapted to animal sounds with curve-like spectral features and our results show that similarity search gives good results in this case. In particular, our method outperforms a general similarity search method based on spectrum projection features and HMMs. Our features are less helpful in the case of noise-like animal sounds which are characteristic for a wide class of species. One future direction of research would

be to incorporate a better description of noise-like sounds in the feature extraction step.

Moreover, an increase in the speed of query processing is highly desirable. One way to achieve this might be to find a criterion how to exclude some of the files from the ranking process by giving fast upper bounds on the ranking value.

REFERENCES

[1] R. Ranft, "Natural sound archives: Past, present and future," *Ann. Brazilian Acad. Sci.*, vol. 76, no. 2, pp. 455–465, 2004.

[2] K.-H. Frommolt, R. Bardeli, F. Kurth, and M. Clausen, "The animal sound archive at the Humboldt-University of Berlin: Current activities in conservation and improving access for bioacoustic research," in *Advances in Bioacoustics 2*, 2006, pp. 139–144.

[3] J. A. Kogan and D. Margoliash, "Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study," *J. Acoust. Soc. Amer.*, vol. 103, no. 4, pp. 2185–2196, Apr. 1998.

[4] H. Mills, "Geographically distributed acoustical monitoring of migrating birds," *J. Acoust. Soc. Amer.*, vol. 108, no. 5, p. 2582, Nov. 2000.

[5] V. B. Deecke, J. K. B. Ford, and P. Spong, "Quantifying complex pattern of bioacoustic variation: Use of a neural network to compare killer whale (orcinus orca) dialects," *J. Acoust. Soc. Amer.*, vol. 105, pp. 2499–2507, 1999.

[6] F. Schwenker, C. Dietrich, H. Kestler, K. Riede, and G. Palm, "Radial basis function neural networks and temporal fusion for the classification of bio acoustic time series," *Neurocomputing*, vol. 51, pp. 265–275, 2003.

[7] P. Somervuo and A. Härmä, "Analyzing bird song syllables on the self-organizing map," in *Workshop on Self-Organizing Maps (WSOM'03)*, Sep. 2003.

[8] N. Mitsakakis, R. Fisher, and A. Walker, "Classification of whale song units using a self-organizing feature mapping algorithm," *J. Acoust. Soc. Amer.*, vol. 100, no. 4, p. 2644, Oct. 1996.

[9] A. Taylor, "Recognising biological sounds using machine learning," in *Proc. Eighth Aust. Joint Conf. Artificial Intelligence*, Nov. 1995, pp. 209–212.

[10] A. Härmä, "Automatic recognition of bird species based on sinusoidal modeling of syllables," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Apr. 2003, vol. 5, pp. 545–548.

[11] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, 1996.

[12] S. Dixon and G. Widmer, "Match: A music alignment tool chest," in *Proc. 6th Int. Conf. Music Information Retrieval*, 2005, pp. 492–491.

[13] F. Kurth and M. Müller, "Efficient index-based audio matching," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 16, no. 2, pp. 382–395, Feb. 2008.

[14] G. Guo and S. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Trans. Neural Netw.*, vol. 14, pp. 209–215, Jan. 2003.

[15] M. Casey, "MPEG-7 sound-recognition tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 737–747, Jun. 2001.

[16] H.-G. Kim, N. Moreau, and T. Sikora, "Audio classification based on mpeg-7 spectral basis representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 716–725, May 2004.

[17] D. Mitrovic, M. Zeppelzauer, and C. Breiteneder, "Discrimination and retrieval of animal sounds," in *12th Int. Multi-Media Modelling Conf. Proc.*, 2006.

[18] D. Ellis, "Classifying music audio with timbral and chroma features," in *Proc. 8th Int. Conf. Music Information Retrieval*, Oct. 2007, pp. 339–340.

[19] M. Müller, F. Kurth, and M. Clausen, "Chroma-based statistical audio features for audio matching," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2005, pp. 275–278.

[20] M. Clausen and F. Kurth, "A unified approach to content-based and fault tolerant music recognition," *IEEE Trans. Multimedia*, vol. 6, no. 5, pp. 717–731, Oct. 2004.

[21] E. Allamanche, J. Herre, B. Fröba, and M. Cremer, "AudioID: Towards content-based identification of audio material," in *Proc. II Oth AES Conv.*, Amsterdam, NL, 2001.

[22] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, "Robust sound modeling for sound identification in broadcast audio," in *Proc. 112th AES Conv.*, Munich, Germany, 2002.

[23] D. Ellis and G. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *Proc. IEEE ICASSP*, 2007, pp. IV-1429–IV-1432.

[24] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic genre classification of music content: A survey," *IEEE Signal Process. Mag.*, vol. 23, pp. 133–141, Mar. 2006.

[25] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *Comput. Music J.*, vol. 28, no. 2, pp. 63–76, 2004.

[26] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Proc. ISPRS Intercommission Conf. Fast Processing of Photogrammetric Data*, June 1987, pp. 281–305.

[27] M. Clausen and F. Kurth, , J. Byrnes, Ed., "Content-based information retrieval by group theoretical methods," in *Computational Noncommutative Algebra and Applications, NATO Science Series, II. Mathematics, Physics and Chemistry*. Norwell, MA: Kluwer, 2004, vol. 136, pp. 29–55.

Rolf Bardeli received the M.S. and Doctor of Natural Sciences (Dr. rer. nat.) degrees in computer science from the University of Bonn, Bonn, Germany, in 2003 and 2008, respectively.

From 2003 to 2008 he was with the Multimedia Signal Processing Group, Department of Computer Science, University of Bonn. Currently, he is with the Competence Center NetMedia of Fraunhofer IAIS, St. Augustin, Germany. His research interests include audio signal processing, multimedia retrieval, algorithmic bioacoustics, complexity theory, and computer algebra.