AUDIO MATCHING VIA CHROMA-BASED STATISTICAL FEATURES

Meinard Müller **Frank Kurth** **Michael Clausen**
Universität Bonn, Institut für Informatik III
Römerstr. 164, D-53117 Bonn, Germany
{meinard, frank, clausen}@cs.uni-bonn.de

ABSTRACT

In this paper, we describe an efficient method for audio matching which performs effectively for a wide range of classical music. The basic goal of audio matching can be described as follows: consider an audio database containing several CD recordings for one and the same piece of music interpreted by various musicians. Then, given a short query audio clip of one interpretation, the goal is to automatically retrieve the corresponding excerpts from the other interpretations. To solve this problem, we introduce a new type of chroma-based audio feature that strongly correlates to the harmonic progression of the audio signal. Our feature shows a high degree of robustness to variations in parameters such as dynamics, timbre, articulation, and local tempo deviations. As another contribution, we describe a robust matching procedure, which allows to handle global tempo variations. Finally, we give a detailed account on our experiments, which have been carried out on a database of more than 110 hours of audio comprising a wide range of classical music.

Keywords: audio matching, chroma feature, music identification

1 INTRODUCTION

Content-based document analysis and retrieval for music data has been a challenging research field for many years now. In the retrieval context, the query-by-example paradigm has attracted a large amount of attention: given a query in form of a music excerpt, the task is to automatically retrieve all excerpts from the database containing parts or aspects similar to the query. This problem is particularly difficult for digital waveform-based audio data such as CD recordings. Due to the complexity of such data, the notion of "similarity" used to compare different audio clips is a delicate issue and largely depends on the respective application as well as the user requirements.

In this paper, we consider the subproblem of *audio matching*. Here the goal is to retrieve all audio clips from the database that in some sense represent the same musical content as the query clip. This is typically the case when the same piece of music is available in several interpretations and arrangements. For example, given a twenty-second excerpt of Bernstein's interpretation of the theme of Beethoven's Fifth, the goal is to find all other corresponding audio clips in the database; this includes the repetition in the exposition or in the recapitulation within the same interpretation as well as the corresponding excerpts in all recordings of the same piece interpreted by other conductors such as Karajan or Sawallisch. It is even more challenging to also include arrangements such as Liszt's piano transcription of Beethoven's Fifth or a synthesized version of a corresponding MIDI file. Obviously, the degree of difficulty increases with the degree of variations one wants to permit in the audio matching.

A straightforward, general strategy for audio matching works as follows: first convert the query as well as the audio files of the database into sequences of suitable audio features. Then compare the feature sequence obtained from the query with feature subsequences obtained from the audio files by means of some suitably defined distance measure. To implement such a procedure, one has to account for the following fundamental questions. Which kind of music is to be considered? What is the underlying notion of similarity to be used in the audio matching? How can this notion of similarity be incorporated in the features and the distance measure? What are typical query lengths? Furthermore, in view of large data sets, the question of efficiency also is of fundamental importance.

Our approach to audio matching follows these lines and works for Western tonal music based on the 12 pitch classes also known as *chroma*. Given a query clip between 10 and 30 seconds of length, the goal in our retrieval scenario is to find all corresponding audio clips regardless of the specific interpretation and instrumentation as described in the above Beethoven example. In other words, the retrieval process has to be robust to changes of parameters such as timbre, dynamics, articulation, and tempo. To this end, we introduce a new kind of audio feature considering short-time statistics over chroma-based energy distributions (see Sect. 3). It turns out that such features are capable of absorbing variations in the aforementioned parameters but are still valuable to distinguish musically un-

related audio clips. The crucial point is that incorporating a large degree of robustness into the audio features allows us to use a relatively rigid distance measure to compare the resulting feature sequences. This leads to robust as well as efficient matching algorithms, see Sect. 4. There, we also explain how to handle global tempo variations by independently processing suitable modifications of the query clip. We evaluated our matching procedure on a database containing more than 110 hours of audio material, which consists of a wide range of classical music and includes complex orchestral and vocal works. In Sect. 5, we will report on our experimental results. Further material and audio examples can be found at `www-mmdb.iai.uni-bonn.de/projects/audiomatching`. In Sect. 2, we give a brief overview of related work and conclude in Sect. 6 with some comments on future work and possible extensions of the audio matching scenario.

2 RELATED WORK

The problem of audio matching can be regarded as an extension of the *audio identification* problem. Here, a query typically consists of short audio fragment obtained from some unknown audio recording. Then the goal is to identify the original recording contained in a given large audio database. Furthermore, the exact position of the query within this recording is to be specified. The identification problem can be regarded as a largely solved problem, even in the presence of noise and slight temporal distortions of the query, see, e.g., Allamanche et al. (2001); Kurth et al. (2002); Wang (2003) and the references therein. Current identification systems, however, are not suitable for a less strict notion of similarity.

In the related problem of *music synchronization*, which is sometimes also referred to as audio matching, one major goal is to align audio recordings of music to symbolic score or MIDI information. One possible approach, as suggested by Turetsky and Ellis (2003) or Hu et al. (2003), is to solve the problem in the audio domain by converting the score or MIDI information into a sequence of acoustic features (e.g., spectral, chroma or MFCC vectors). By means of dynamic time warping, this sequence is then compared with the corresponding feature sequence extracted from the audio version. Note that the objective of our audio matching scenario is beyond the one of audio synchronization: in the latter case the goal is to time-align two given versions of the same underlying piece of music, whereas in the audio matching scenario the goal is to identify short audio fragments similar to the query hidden in the database.

The design of audio features that are robust to variations of specific parameters is of fundamental importance to most content-based audio analysis applications. Among a large number of publications, we quote two papers representing different strategies, which will be applied in our feature design. The *chroma-based approach* as suggested by Bartsch and Wakefield (2005) represents the spectral energy contained in each of the 12 traditional pitch classes of the equal-tempered scale. Such features strongly correlate to the harmonic progression of the audio, which are often prominent in Western music. Another general strategy is to consider certain *statistics* such as pitch histograms for audio signals, which may suffice to distinguish different music genre, see, e.g., Tzanetakis et al. (2002). We will combine aspects of these two approaches in evaluating chroma-based audio features by means of short-time statistics.

3 AUDIO FEATURES

In this section, we give a detailed account on the design of audio features, possessing a high degree of robustness to variations of parameters such as timbre, dynamics, articulation, and local tempo deviations as well as to slight variations in note groups such as trills or grace notes. Correlating strongly to the harmonics information contained in the audio signals, the features are well suited for our audio matching scenario. In the feature design, we proceed in two-stages: in the first stage, we use a small analysis window to investigate how the signal's energy locally distributes among the 12 chroma classes (Sect. 3.1). In the second stage, we use a much larger (concerning the actual time span measured in seconds) statistics window to compute thresholded short-time statistics over these energy distributions (Sect. 3.2). In Sect. 3.3, we then discuss the qualities as well as drawbacks of the resulting features.

3.1 Chroma Feature

The local chroma energy distributions (first stage) are computed as follows.

(1) Decompose the audio signal into 88 frequency bands corresponding to the musical notes A0 to C8 (MIDI pitches $p = 21$ to $p = 108$). To properly separate adjacent notes, we use a filter bank consisting of elliptic filters with excellent cut-off properties as well as the forward-backward filtering strategy as described by Müller et al. (2004).
(2) Compute the short-time mean-square power (STMSP) for each of the 88 subbands by convolving the squared subband signals with a rectangular window corresponding to 200 ms with an overlap of half the size.
(3) Compute STMSPs of all chroma classes by adding up the corresponding STMSPs of all pitches belonging to the respective class. For example, to compute the STMSP of the chroma class A, add up the STMSPs of the pitches A0,A1,...,A7. This yields a real 12-dimensional vector $\vec{v} = (v_1, \ldots, v_{12}) \in \mathbb{R}^{12}$ for each analysis window.
(4) Finally, for each window compute the energy distribution relative to the 12 chroma classes by replacing the vectors $\vec{v}$ from Step (3) by $\vec{v}/(\sum_{i=1}^{12} v_i)$.

Altogether, the audio signal is converted into a sequence of 12-dimensional chroma distribution vectors—10 vectors per second, each vector corresponding to 200 ms. For the Beethoven example, the resulting 12 curves are shown in Fig. 1. To suppress random-like energy distributions occurring during passages of extremely low energy, (e.g., passages of silence before the actual start of the recording or during long pauses), we assign an equally distributed chroma energy to these passages.

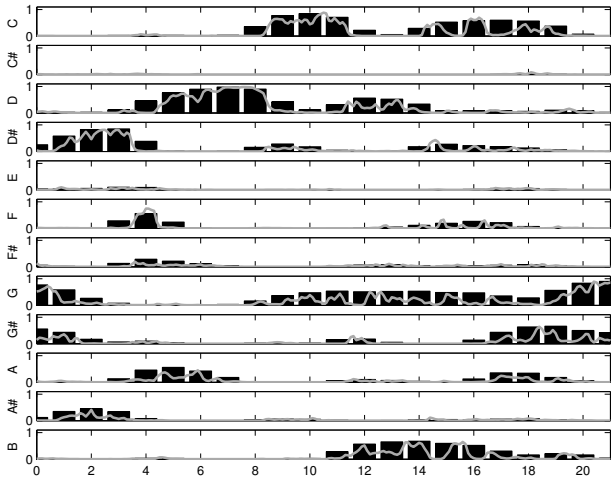

Figure 1: The first 21 seconds (first 20 measures) of Bernstein's interpretation of Beethoven's Fifth Symphony. The light curves represent the local chroma energy distributions (10 features per second). The dark bars represent the CENS features (1 feature per second).

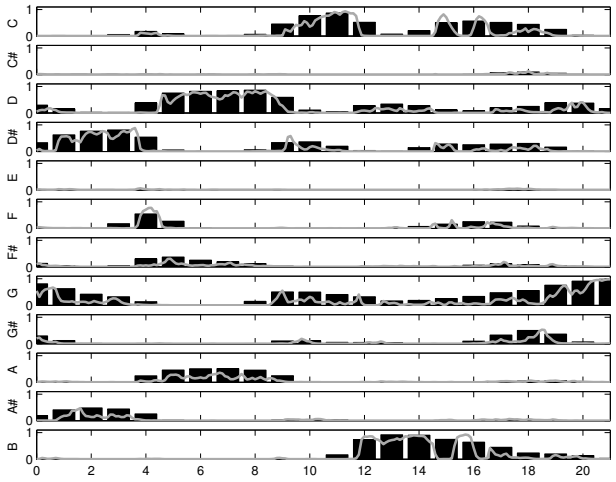

Figure 2: CENS features for the first 21 seconds of Sawallisch's recording corresponding to the same measures as the Beethoven example of Fig. 1.

3.2 Short-time statistics

In view of our audio matching application, the local chroma energy distribution features are still too sensitive, particularly when looking at variations in the articulation and local tempo deviations. Therefore, we introduce a second, much larger statistics window and consider suitable statistics concerning the energy distributions over this window. The details of the second stage are as follows:

(5) Quantize each normalized chroma vector $\vec{v} = (v_1, \ldots, v_{12})$ from Step (4) by assigning the value 4 if a chroma component v_i exceeds the value 0.4 (i.e., if it contains more than 40 percent of the signal's total energy in the ith chroma component for the respective analysis window). Similarly, we assign the value 3 if $0.2 \leq v_i < 0.4$, the value 2 if $0.1 \leq v_i < 0.2$, the value 1 if $0.05 \leq v_i < 0.1$, and the value 0 otherwise. For example, the chroma vector $\vec{v} = (0.02, 0.5, 0.3, 0.07, 1.1, 0, \ldots, 0)$ is thus transformed into the vector $\vec{v}^q := (0, 4, 3, 1, 2, 0, \ldots, 0)$.

(6) Convolve the sequence of the quantized chroma vectors from Step (5) component-wise using a Hann window of length 41. This again results in a sequence of 12-dimensional vectors with non-negative entries, representing a kind of weighted statistics of the energy distribution over a window of 41 consecutive chroma vectors. In a last step, downsample the sequence by a factor of 10 and normalize the vectors with respect to the Euclidean norm.

Thus, after Step (6) we obtain one vector per second, each spanning roughly 4100 ms of audio. For short, these features are simply referred to as *CENS features* (**C**hroma **E**nergy distribution **N**ormalized **S**tatistics), which are elements of the set $\mathcal{F}$ of vectors defined by $\mathcal{F} := \left\{\vec{x} = (x_1, \ldots, x_{12}) \in \mathbb{R}^{12} \mid x_i \geq 0, \sum_{i=1}^{12} x_i^2 = 1\right\}$. Fig. 1 shows the resulting sequence of CENS features for our running example.

3.3 Discussion of CENS features

As mentioned above, the CENS feature sequences correlate closely with the smoothed harmonic progression of the underlying audio signal. Such sequences, as illustrated by Fig. 1 and Fig. 2, often characterize a piece of music accurately but independently of the specific interpretation. Other parameters, however, such as dynamics, timbre, or articulation are masked out to a large extent: the normalization in Step (4) makes the CENS features invariant to dynamic variations. Furthermore, using chroma instead of pitches (see Step (3)) not only takes into account the close octave relationship in both melody and harmony as typical for Western music (see Bartsch and Wakefield (2005)), but also introduces a high degree of robustness to variations in timbre. Then, applying energy thresholds (see Step (5)) makes the CENS features insensitive to noise components as may arise during note attacks. Finally, taking statistics over relatively large windows not only smoothes out local time deviations as may occur for articulatory reasons but also compensates for different realizations of note groups such as trills or arpeggios.

A major problem with the feature design is to satisfy two conflicting goals: robustness on the one hand and accuracy on the other hand. Our two-stage approach admits a high degree of flexibility in the feature design to find a good tradeoff. The small window in the first stage is used to pick up local information, which is then statistically evaluated in the second stage with respect to a much larger window—note that simply enlarging the analysis window in Step (2) without using the second stage may average out valuable local harmonics information leading to less meaningful features. Furthermore, modifying parameters of the second stage such as the size of the statistics window or the thresholds in Step (5) allows to enhance or mask out certain aspects without repeating the cost-intensive computations in the first stage. We will make use of this strategy in Sect. 4.2, when dealing with the problem of global tempo variations.

Finally, we want to mention some problems concerning CENS features. The usage of a filter bank with fixed

frequency bands is based on the assumption of well-tuned instruments. Slight deviations of up to 30–40 cents from the center frequencies can be tackled by the filters, which have relatively wide pass bands of constant amplitude response. Global deviations in tuning can be compensated by employing a suitably adjusted filter bank. However, phenomena such as strong string vibratos or pitch oscillation as is typical for, e.g., kettle drums lead to significant and problematic pitch smearing effects. Here, the detection and smoothing of such fluctuations, which is certainly not an easy task, may be necessary prior to the filtering step. However, as we will see in Sect. 5, the CENS features generally still lead to good matching results even in presence of the artifacts mentioned above.

4 AUDIO MATCHING

In this section, we first describe the basic idea of our audio matching procedure, then explain how to incorporate invariance to global tempo variations, and close with some notes on efficiency.

4.1 Basic matching procedure

The audio database consists of a collection of CD audio recordings, typically containing various interpretations for one and the same piece of music. To simplify things, we may assume that this collection is represented by one large document D by concatenating the individual recordings (we keep track of the boundaries in a supplemental data structure). The query Q consists of a short audio clip, typically lasting between 10 and 30 seconds. In the feature extraction step, as described in Sect. 3, the document D as well as the query Q are transformed into sequences of CENS-feature vectors. We denote these feature sequences by $F[D] = (\vec{v}^1, \vec{v}^2, \ldots, \vec{v}^N)$ and $F[Q] = (\vec{w}^1, \vec{w}^2, \ldots, \vec{w}^M)$ with $\vec{v}^n \in \mathcal{F}$ for $n \in [1:N]$ and $\vec{w}^m \in \mathcal{F}$ for $m \in [1:M]$.

The goal of audio matching is to identify audio clips in D that are similar to Q. To this end, we compare the sequence $F[Q]$ to any subsequence of $F[D]$ consisting of M consecutive vectors. More specifically, letting $\vec{X} = (\vec{x}^1, \ldots, \vec{x}^M) \in \mathcal{F}^M$ and $\vec{Y} = (\vec{y}^1, \ldots, \vec{y}^M) \in \mathcal{F}^M$, we set $d^M(\vec{X}, \vec{Y}) := 1 - \frac{1}{M}\sum_{m=1}^{M} \langle \vec{x}^m, \vec{y}^m \rangle$, where $\langle \vec{x}^m, \vec{y}^m \rangle$ denotes the inner product of the vectors $\vec{x}^m$ and $\vec{y}^m$ (thus coinciding with the cosine of the angle between $\vec{x}^m$ and $\vec{y}^m$, since $\vec{x}^m$ and $\vec{y}^m$ are assumed to be normalized). Note that d^M is zero in case $\vec{X}$ and $\vec{Y}$ coincide and assumes values in the real interval $[0,1] \subset \mathbb{R}$. Next, we define the distance function $\Delta : [1:N] \to [0,1]$ with respect to $F[D]$ and $F[Q]$ by

$$\Delta(i) := d^M((\vec{v}^i, \vec{v}^{i+1} \ldots, \vec{v}^{i+M-1}), (\vec{w}^1, \vec{w}^2, \ldots, \vec{w}^M))$$

for $i \in [1 : N - M + 1]$ and $\Delta(i) := 1$ for $i \in [N - M + 2 : N]$. In particular, $\Delta(i)$ describes the distance between $F[Q]$ and the subsequence of $F[D]$ starting at position i and consisting of M consecutive vectors. The computation of Δ is also illustrated by Fig. 3.

We now determine the best matches of Q within D by successively considering minima of the distance function

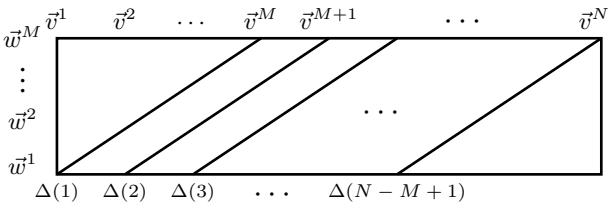

Figure 3: Schematic illustration of the computation of the distance function Δ with respect to $F[Q] = (\vec{w}^1, \ldots, \vec{w}^M)$ and $F[D] = (\vec{v}^1, \ldots, \vec{v}^N)$.

Δ: in the first step, we determine the index $i \in [1:N]$ minimizing Δ. Then the audio clip corresponding to the feature sequence $(\vec{v}^i, \vec{v}^{i+1} \ldots, \vec{v}^{i+M-1})$ is our best match. We then exclude a neighborhood of length M of the best match from further considerations by setting $\Delta(j) = 1$ for $j \in [i - \lceil M/2 \rceil : i + \lceil M/2 \rceil] \cap [1:N]$, thus avoiding matches with a large overlap to the subsequent matches. In the second step, we determine the feature index minimizing the modified distance function, resulting in the second best match, and so on. This procedure is repeated until a predefined number of matches has been retrieved or until the distance of a retrieved match exceeds a specified threshold.

As an illustrating example, let's consider a database D consisting of four pieces: one interpretation of Bach's Toccata BWV565, two interpretations (Bernstein, Sawallisch) of the first movement of Beethoven's Fifth Symphony op. 67, and one interpretation of Shostakovich's Waltz 2 from his second Jazz Suite. The query Q again consists of the first 21 seconds (20 measures) of Bernstein's interpretation of Beethoven's Fifth Symphony (cf. Fig. 1). The upper part of Fig. 4 shows the resulting distance function Δ. The lower part shows the feature sequences corresponding to the ten best matches sorted from left to right according to their distance. Here, the best match (coinciding with the query) is shown on the leftmost side, where the matching rank and the respective Δ-distance ($1/0.011$) are indicated above the feature sequence and the position ($0 - 21$, measured in seconds) within the audio file is indicated below the feature sequence. Corresponding parameters for the other nine matches are given in the same fashion.

Note that the distance 0.011 for the best match is not exactly zero, since the interpretation in D starts with a small segment of silence, which has been removed from the query Q. Furthermore, note that the first 20 measures of Beethoven's Fifth, corresponding to Q, appear again in the repetition of the exposition and once more with some slight modifications in the recapitulation. Matches 1, 2, and 5 correspond to these excerpts in Bernstein's interpretation, whereas matches 3, 4, and 6 to those in Sawallisch's interpretation. In Sect. 5, we continue this discussion and give additional examples.

4.2 Global tempo variations

So far, our matching procedure only considers subsequences of $F[D]$ having the same length M as $F[Q]$. As a consequence, a global tempo difference between two

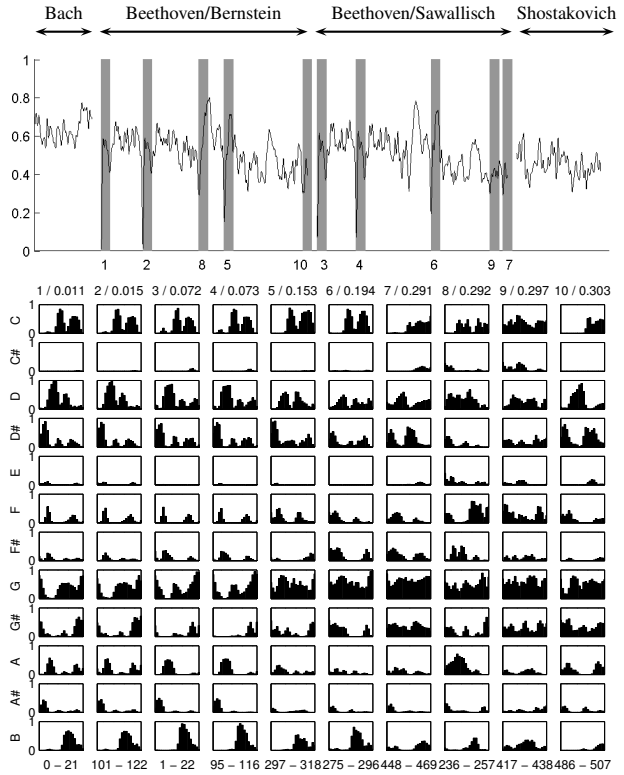

Figure 4: Distance function Δ (top) and CENS feature sequences of the first ten matches for a data set D consisting of four pieces and query Q corresponding to Fig. 1.

audio clips, even though representing the same excerpt of music, will typically lead to a larger distance than it should. For example, Bernstein's interpretation of the first movement of Beethoven's Fifth is much slower (roughly 85 percent) than Karajan's interpretation. While there are 21 CENS feature vectors for the first 20 measures computed from Bernstein's interpretation, there are only 17 in Karajan's case. To account for such global tempo variations in the audio matching scenario, we create several versions of the query audio clip corresponding to different tempos and then process all these query versions independently. Here, our two-stage approach exhibits another benefit, since such tempo changes can be simulated by changing the size of the satistics window as well as the downsampling factor in Steps (5) and (6) of the CENS feature computation. For example, using a window size of 53 (instead of 41) and a downsampling factor of 13 (instead of 10) simulates a tempo change by a factor of $10/13 \approx 0.77$ of the origianl query. In our experiments, we used 8 different query versions as indicated by Table 1, covering global tempo variations of roughly -40 to $+40$ percent.

Next, for each of the eight resulting CENS-feature sequences we compute a distance function denoted by $\Delta^7, \ldots, \Delta^{14}$ (the index indicating the downsampling factor). In particular, the original distance function Δ equals Δ^{10}. Finally, we define $\Delta^{\min} : [1:N] \to [0,1]$ by setting $\Delta^{\min}(i) := \min(\Delta^7(i), \ldots, \Delta^{14}(i))$ for $i \in [1:N]$. We then proceed with $\Delta^{\min}$ as described in Sect 4.1 to determine the best audio matches. Fig. 5 illustrates how

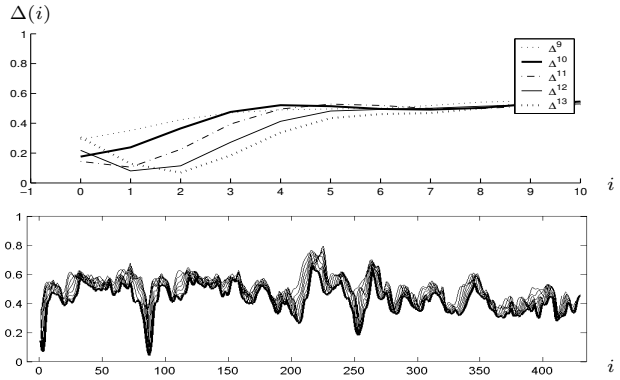

Figure 5: Top: $\Delta^9, \ldots, \Delta^{13}$ (first eleven values) for the 21 second Bernstein query applied to Karajan's interpretation. Bottom: $\Delta^7, \ldots, \Delta^{14}$ and $\Delta^{\min}$-distance function.

ws	29	33	37	**41**	45	49	53	57
df	7	8	9	**10**	11	12	13	14
tc	1.43	1.25	1.1	**1.0**	0.9	0.83	0.77	0.7

Table 1: Tempo changes (tc) simulated by changing statistics window sizes (ws) and downsampling factors (df).

changing the query tempo affects the distance function.

In conclusion, we note that global tempo deviations are accounted for by employing several suitably modified queries, whereas local tempo deviations are absorbed to a high degree by using CENS features.

4.3 Efficient implementation

At this point, we want to mention that the distance function Δ given by $\Delta(i) = 1 - \frac{1}{M}\sum_{m=1}^{M}\langle \vec{v}^{i+m-1}, \vec{w}^m\rangle$ can be computed efficiently. Here, one has to note that each of the 12 components of the vector $\sum_{m=1}^{M}\langle \vec{v}^{i+m-1}, \vec{w}^m\rangle$ can be expressed as a convolution, which can then be evaluated efficiently using FFT-based convolution algorithms. By this technique, Δ can be calculated with $O(DN \log M)$ operations, where $D = 12$ denotes the dimension of the vectors. In other words, the query length M only contributes a logarithmic factor to the total arithmetic complexity. Thus, even long queries may be processed very efficiently. The experimental setting as well as the running time to process a typical query is described in the next section.

5 EXPERIMENTS

We implemented our audio matching procedure in MATLAB and tested it on a database containing 112 hours of uncompressed audio material (mono, 22050 Hz), requiring 16.5 GB of disk space. The database comprises 1167 audio files reflecting a wide range of classical music, including, among others, pieces by Bach, Bartok, Bernstein, Beethoven, Chopin, Dvorak, Elgar, Mozart, Orff, Ravel, Schubert, Shostakovich, Vivaldi, and Wagner. In particular, it contains all Beethoven symphonies, all Beethoven piano sonatas, all Mozart piano concertos, several Schubert and Dvorak symphonies—many of the

pieces in several versions. Some of the orchestral pieces are also included as piano arrangements or synthesized MIDI-versions. In a preprocessing step, we computed the CENS features for all audio files of the database, resulting in a single sequence $F[D]$ as described in Sect. 4.1. Storing the features $F[D]$ requires only 40.3 MB (opposed to 16.5 GB for the original data), amounting in a data reduction of a factor of more than 400. Note that the feature sequence $F[D]$ is all we need during the matching procedure. Our tests were run on an Intel Pentium IV, 3 GHz with 1 GByte RAM under Windows 2000. Processing a query of 10 to 30 seconds of duration takes roughly one second w.r.t. Δ and about $7-10$ seconds w.r.t. $\Delta^{\min}$. As is also mentioned in Sect. 6, the processing time may further be reduced by employing suitable indexing methods.

5.1 Representative matching results

We now discuss in detail some representative matching results obtained from our procedure, using the query clips shown in Table 2. For each query clip, the columns contain from left to right an acronym, the specification of the piece of music, the measures corresponding to the clip, and the interpreter. Demo audio material of the examples discussed in this paper is provided at `www-mmdb.iai.uni-bonn.de/projects/audiomatching`, where additional matching results and visualizations can be found as well.

We continue our Beethoven example. Recall that the query, in the following referred to as "BeetF" (see Table 2), corresponds to the first 20 measures, which appear once more in the repetition of the exposition and with some slight modifications in the recapitulation. Since our database contains Beethoven's Fifth in five different versions—four orchestral version conducted by Bernstein, Karajan, Kegel, and Sawallisch, respectively, and Liszt's piano transcription played by Scherbakov—there are altogether 15 occurrences in our database similar to the query "BeetF". Using our matching procedure, we automatically determined the best 15 matches in the entire database w.r.t. $\Delta^{\min}$. Those 15 matches contained 14 of the 15 "correct" occurences—only the 14th match (distance 0.217) corresponding to some excerpt of Schumann's third symphony was "wrong". Furthermore, it turned out that the first 13 matches are exactly the ones having a $\Delta^{\min}$-distance of less than 0.2 from the query, see also Fig. 6 and Table 3. The 15th match (excerpt in the recapitulation by Kegel) already has a distance of 0.220. Note that even the occurrences in the exposition of Scherbakov's piano version were correctly identified as 11th and 13th match, even though differing significantly in timbre and articulation from the orchestral query. Only the occurrence in the recapitulation of the piano version was not among the top matches.

As a second example, we queried the piano version "BeLiF" of about 26 seconds of duration (see Table 2), which corresponds to the first part of the development of Beethoven's Fifth. The $\Delta^{\min}$-distances of the best twenty matches are shown in Table 3. The first six of these matches contain all five "correct" occurrences in the five interpretations corresponding to the query excerpt, see also Fig 7. Only the 4th match comes from the first movement (measures 200–214) of Mozart's symphony No. 40, KV 550. Even though seemingly unrelated to the query, the harmonic progression of Mozart's piece exhibits a strong correlation to the Beethoven query at these measures. As a general tendency, it has turned out in our experiments that for queries of about 20 seconds of duration the "correct" matches have a distance lower than 0.2 to the query. In general, only few "false" matches have a $\Delta^{\min}$-distance to the query lower than this distance threshold.

A similar result was obtained when querying "SchuU" corresponding to measures 9–21 of the first theme of Schubert's "Unfinished" conducted by Abbado. Our database contains the "Unfinished" in six different interpretations (Abbado, Maag, Mik, Nanut, Sacci, Solti), the theme appearing once more in the repetition of the exposition and in the recapitulation. Only in the Maag interpreta-

Query	Piece	measures	interpreter
BachAn	Bach BWV 988, Goldberg "Aria"	1-n	MIDI
BeetF	Beethoven Op. 67 "Fifth"	1-20	Bernstein
BeLiF	Beethoven Op. 67 "Fifth" (Liszt)	129-170	Scherbakov
Orff	Carmina Burana	1-4	Jochum
SchuU	Schubert D759 "Unfinished"	9-21	Abbado
ShoWn	Shostakovich Jazz Suite 2, Waltz 2	1-n	Chailly
VivaS	RV269 No.1 "Spring"	44-55	MIDI

Table 2: Query audio clips used in the experiments. If not specified otherwise, the measures correspond to the first movement of the respective piece.

No.	BachA8	BeetF	BeLiF	Orff	ShoW22	SchuU	VivaS
1	0.005	0.011	0.010	0.005	0.017	0.024	0.095
2	0.020	0.015	0.139	0.037	0.051	0.052	0.139
3	0.090	0.044	0.142	0.065	0.098	0.061	0.154
4	0.093	0.051	0.168	0.138	0.104	0.070	0.155
5	0.093	0.058	0.168	0.148	0.109	0.071	0.172
6	0.095	0.069	0.172	0.150	0.140	0.072	0.210
7	0.098	0.072	0.200	0.152	0.148	0.073	0.221
8	0.102	0.073	0.203	0.155	0.163	0.091	0.238
9	0.104	0.143	0.204	0.158	0.167	0.097	0.241
10	0.107	0.180	0.214	0.165	0.173	0.100	0.244
11	0.107	0.183	0.221	0.166	0.186	0.101	0.248
12	0.108	0.195	0.221	0.166	0.187	0.103	0.257
13	0.110	0.197	0.225	0.167	0.188	0.107	0.262
14	0.110	0.217	0.229	0.179	0.192	0.108	0.267
15	0.112	0.220	0.230	0.179	0.193	0.122	0.268
16	0.114	0.224	0.231	0.172	0.194	0.151	0.271
17	0.117	0.225	0.232	0.173	0.197	0.158	0.273
18	0.120	0.229	0.234	0.174	0.198	0.205	0.275
19	0.122	0.237	0.235	0.174	0.199	0.207	0.276
20	0.122	0.238	0.236	0.176	0.199	0.214	0.279

Table 3: Each column shows the $\Delta^{\min}$-distances of the twenty best matches to the query indicated by Table 2.

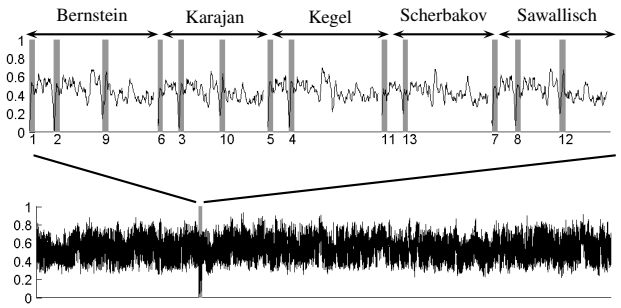

Figure 6: Bottom: $\Delta^{\min}$-distance function for the entire database w.r.t. the query "BeetF". Top: Enlargement showing the five interpretations of the first movement of Beethoven's Fifth containing all of the 13 matches with $\Delta^{\min}$-distance < 0.2 to the query.

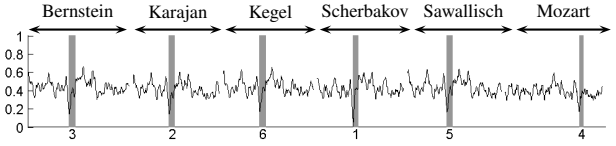

Figure 7: Section consisting of the five interpretations of the first movement of Beethoven's Fifth and the first movement of Mozart's symphony No. 40, KV 550. The five occurences in the Beethoven interpretations are among the best six matches, all having $\Delta^{\min}$-distance < 0.2 to the query "BeLiF".

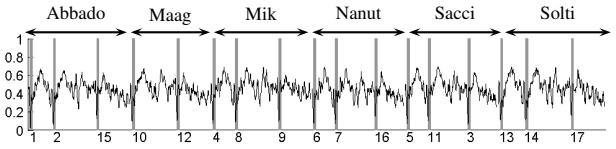

Figure 8: Section consisting of the five interpretations of the first movement of Schubert's Unfinished. The 17 occurences exactly correspond to the 17 matches with $\Delta^{\min}$-distance < 0.2 to the query "SchuU".

tion the exposition is not repeated, leading to a total number of 17 occurrences similar to the query. The best 17 matches retrieved by our algorithm exactly correspond to these 17 occurences, all of those matches having a $\Delta^{\min}$-distance well below 0.2, see Table 3 and Fig. 8. The 18th match, corresponding to some excerpt of Chopin's Scherzo Op. 20, already had a $\Delta^{\min}$-distance of 0.205.

Our database also contains two interpretations (Jochum, Ormandy) of the Carmina Burana by Carl Orff, a piece consisting of 25 short episodes. Here, the first episode "O Fortuna" appears again at the end of the piece as 25th episode. The query "Orff" corresponds to the first four measures of "O Fortuna" in the Jochum interpretation (22 seconds of duration), employing the full orchestra, percussion, and chorus. Again, the best four matches exactly correspond to the first four measures in the first and 25th episodes of the two interpretations. The fifth match is then an excerpt from the third movement of Schumann's Symphony No. 4, Op. 120. When asking for all matches having a $\Delta^{\min}$-distance of less than 0.2 to the query, our matching procedure retrieved 75 matches from the database. The reason for the relatively large number of matches within a small distance to the query is the relatively unspecific, unvaried progression in the CENS-feature sequence of the query, which is shared by many other pieces as well. In Sect. 5.2, we will discuss a similar example ("BachAn") in more detail. It is interesting to note that among the 75 matches, there are 22 matches from various episodes of the Carmina Burana, which are variations of the original theme.

To test the robustness of our matching procedure to the respective instrumentation and articulation, we also used queries synthesized from uninterpreted MIDI versions. For example, the query "VivaS" (see Table 2) consists of a synthesized version of the measures 44–55 of Vivaldi's Spring RV269, No. 1. This piece is contained in our database in 7 different interpretations. The best seven matches were exactly the "correct" excerpts, where

query	ShoW12	ShoW20	ShoW27
duration (sec)	13	22	29
#(matches, $\Delta^{\min} \leq 0.2$)	590	23	8
Chailly	1/2/6/10	1/2/7/3	1/2/7/4
Yablonsky	119/59/103/138	4/5/36/6	3/5/8/6

Table 4: Total number of matches with $\Delta^{\min}$-distance lower than 0.2 for queries of different durations.

the first 5 of these matches had a $\Delta^{\min}$-distance of less than 0.2 from the query (see also Table 3). The robustness to different instrumentations is also shown by the Shostakovich example in the next section.

5.2 Dependence on query length

Not surprisingly, the quality of the matching results depends on the length of the query: queries of short duration will generally lead to a large number of matches in a close neighborhood of the query. Enlarging the query length will generally reduce the number of such matches. We illustrate this principle by means of the second Waltz of Shostakovich's Jazz Suite No. 2. This piece is of the form $A_1A_2BA_3A_4$, where the first theme consists of 38 measures and appears four times (parts A_1, A_2, A_3, A_4), each time in a different instrumentation. In part A_1 the melody is played by strings, then in A_2 by clarinet and wood instruments, in A_3 by trombone and brass, and finally in A_4 in a tutti version. The Waltz is contained in our database in two different interpretations (Chailly,Yablonsky) leading to a total number of 8 occurrences of the theme.

The query "ShoWn" (see Table 2) consists of the first n measures of the theme in the Chailly interpretation. Table 4 compares the total number of matches to the query duration. For example, the query clip "ShoW12" (duration of 13 seconds) leads to 590 matches with a $\Delta^{\min}$-distance lower than 0.2. Among these matches the four occurrences A_1, A_2, A_3, and A_4 in the Chailly interpretation could be found at position 1 (the query itself), 2, 6 and 10, respectively. Similarly, the four occurrences in the Yablonsky interpretation could be found at the positions $119/59/103/138$. Enlarging the query to 20 measures (22 seconds) led to a much smaller number of 23 matches with a $\Delta^{\min}$-distance lower than 0.2. Only the trombone theme in the Yablonsky version (36th match with $\Delta^{\min}$-distance of 0.207) was not among the first 23 matches. Finally, querying "ShoW27" led to 8 matches with a $\Delta^{\min}$-distance lower than 0.2, exactly corresponding to the eight "correct" occurrences, see Fig. 9. Among these matches, the two trombone versions have the largest $\Delta^{\min}$-distances. This is caused by the fact that the spectra of low-pitched instruments such as the trombone generally exhibit phenomena such as oscillations and smearing effects resulting in degraded CENS features.

As a final example, we consider the Goldberg Variations by J.S. Bach, BWV 988. This piece consists of an Aria, thirty variations and a repetition of the Aria at the end of the piece. The interesting fact is that the variations are on the Aria's bass line, which closely correlates with the harmonic progression of the piece. Since the sequence of CENS features also closely correlates with this progression, a large number of matches is to be expected when querying the theme of the Aria. The query

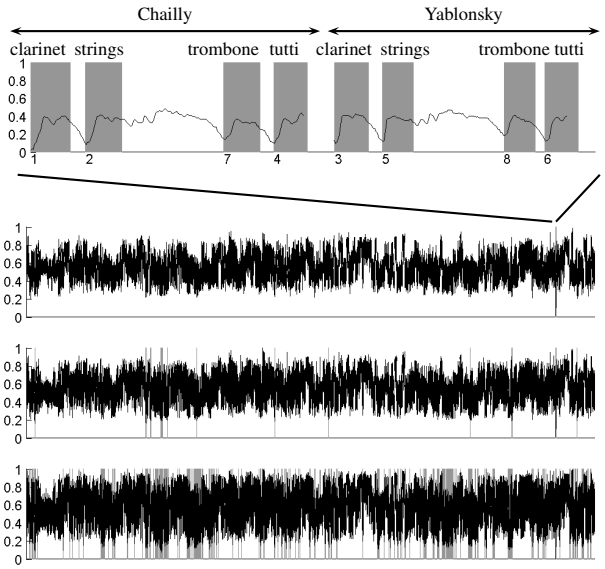

Figure 9: Second to fourth row: $\Delta^{\min}$-distance function for the entire database w.r.t. the queries "ShoW27", "ShoW20", and "ShoW12". The light bars indicate the matching regions. First row: Enlargement for the query "ShoW27" showing the two interpretations of the Waltz. Note that the theme appears in each interpretation in four different instrumentations.

"BachAn" consists of the first n measures of the Aria synthesized from some uninterpreted MIDI, see Table 2. Querying "BachA4" (10 seconds of duration) led to 576 matches with $\Delta^{\min}$-distance of less than 0.2. Among these matches, 214 correspond to some excerpt originating from a variation of one of the four Goldberg interpretations contained in our database. Increasing the duration of the query, we obtained 307 such matches for "BachA8" (20 seconds), 195 of them corresponding to some Goldberg excerpt. Similarly, one obtained 144 such matches for "BachA12" (30 seconds), 127 of them corresponding to some Goldberg excerpt.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced an audio matching procedure which, given a query audio clip of between 10 and 30 seconds of duration, automatically and efficiently identifies all corresponding audio clips in the database irrespective of the specific interpretation or instrumentation. A representative selection of our experimental results, including the ones discussed in this paper, can be found at `www-mmdb.iai.uni-bonn.de/projects/audiomatching`. As it turns out, our procedure performs well for most of our query examples within a wide range of classical music proving the usefulness of our CESN features. The top matches almost always include the "correct" occurrences, even in case of synthesized MIDI versions and interpretations in different instrumentations.

In conclusion, our experimental results suggest that a query duration of roughly 20 seconds seems to be sufficient for a good characterization of most audio excerpts. Enlarging the duration generally makes the matching process even more stable and reduces the number of "false" matches. Our matching process may produce a large number of "false" matches (false positives) or miss "correct" matches (false negatives) in case the underlying music does not exhibit characteristic harmonics information, as is, for example, the case for music with an unchanging harmonic progression or for purely percussive music. "False" matches with small $\Delta^{\min}$-distance generally differ considerably from the query (accidentally having a similar harmonic progression). Here, our future goal is to provide the user with a choice of additional, orthogonal features such as beat, timbre, or dynamics, to allow for a ranking adapted to the user's needs.

For the future, we also plan to employ indexing methods to significantly reduce the query times of our matching algorithm (in the present implementation it requires 7–10 seconds for processing single query w.r.t. $\Delta^{\min}$). As a further extension of our matching procedure, we also want to retrieve audio clips that differ from the query by a global pitch transposition. This, e.g., includes arrangements played in different keys or themes appearing in various keys as is typically the case for a sonata. First experiments show that such pitch transpositions can be handled by cyclically shifting the components of the CENS features extracted from the query.

As an application, we plan to employ our audio matching strategy to substantially accelerate music synchronization. Here, the idea is to identify salient audio matches, which can then be used as anchor matches as suggested by Müller et al. (2004).

Finally, note that we evaluated our experiments manually, by comparing the retrieved matches with the expected occurrences as a ground truth (knowing exactly the configuration of our audio database). Here, an automated procedure allowing to conduct large-scale tests is an important issue to be considered.

REFERENCES

E. Allamanche, J. Herre, B. Fröba, and M. Cremer. AudioID: Towards Content-Based Identification of Audio Material. In *Proc. 110th AES Convention, Amsterdam, NL*, 2001.

M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Trans. on Multimedia*, 7(1):96–104, Feb. 2005.

N. Hu, R. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE WASPAA, New Paltz, NY*, October 2003.

F. Kurth, M. Clausen, and A. Ribbrock. Identification of highly distorted audio material for querying large scale data bases, 2002.

M. Müller, F. Kurth, and T. Röder. Towards an efficient algorithm for automatic score-to-audio synchronization. In *Proc. ISMIR, Barcelona, Spain*, 2004.

R. J. Turetsky and D. P. Ellis. Force-Aligning MIDI Syntheses for Polyphonic Music Transcription Generation. In *Proc. ISMIR, Baltimore, USA*, 2003.

G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. In *Proc. ISMIR, Paris, France*, 2002.

A. Wang. An Industrial Strength Audio Search Algorithm. In *Proc. ISMIR, Baltimore, USA*, 2003.