CHROMA-BASED STATISTICAL AUDIO FEATURES FOR AUDIO MATCHING

Meinard Müller *Frank Kurth* *Michael Clausen*

Department of Computer Science III, University of Bonn
Römerstr. 164, D-53117 Bonn, Germany
{meinard, frank, clausen}@cs.uni-bonn.de

ABSTRACT

Large music collections often contain several recordings of the same piece of music, which are interpreted by various musicians and possibly arranged in different instrumentations. Given a short query audio clip, an important task in audio retrieval is to automatically and efficiently identify all corresponding audio clips irrespective of the specific interpretation or instrumentation. In view of this problem, which is also referred to as audio matching, the main contribution of this paper is to introduce a new type of audio feature that strongly correlates to the harmonic progression of the audio signal. In addition, our feature shows a high degree of robustness to variations in parameters such as dynamics, timbre, articulation, and local tempo deviations. The feature design is carried out in two stages basically taking short-time statistics over chroma-based energy distributions. Here, the chroma correspond to the 12 traditional pitch classes of the equal-tempered scale. Applied to audio matching on a large audio database consisting of a wide range of classical music (112 hours of audio material), our features proved to be a powerful tool providing accurate matchings in an efficient way concerning time as well as memory requirements.

1. INTRODUCTION

Content-based document analysis and efficient audio browsing in large music databases has become an important issue in music information retrieval. Here, the automatic annotation of audio data by descriptive high-level features as well as the automatic generation of cross-links between audio excerpts of similar musical content are of major concern. In this paper, we address the subproblem of *audio matching*. To illustrate this problem, we consider an audio database containing several CD recordings for one and the same piece of music, which is interpreted by various musicians. For example, Vivaldi's "Four Seasons" may be available in the interpretation by Pinchas Zukerman, Itzhak Perlman, and Vanessa Mae. Then, given a twenty-second excerpt of Zukerman's interpretation of the violin solo in the "Spring", the goal is to automatically retrieve the corresponding excerpts in the other interpretations. It is even more challenging to also include different arrangements of the same piece such as a piano transcription or a synthesized MIDI version. Obviously, the degree of difficulty increases with the degree of variations one wants to permit in the audio matching.

In our matching scenario the goal is, given a query audio clip of between 10 and 30 seconds of duration, to find all corresponding audio clips regardless of the specific interpretation and instrumentation as described in the above Vivaldi example. In other words, the retrieval process has to be robust to parameters such as timbre, dynamics, articulation, and tempo. The main contribution of this paper is to introduce a new type of audio feature that takes short-time statistics over chroma-based energy distributions, see Sect. 2. It turns out that such features are capable of absorbing variations in the aforementioned parameters but are still valuable to distinguish musically unrelated 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, see Sect. 3. This not only allows us to design very efficient matching algorithms but also to incorporate further degrees of freedom concerning the global tempo and global pitch transpositions. Our experimental results show the matching accuracy of our features, see Sect. 3 and `www-mmdb.iai.uni-bonn.de/projects/audiomatching`.

The problem of audio matching can be regarded as an extension of the *audio identification* problem. Here, given a query consisting of short audio clip, the goal is to identify the original recording hidden in some large audio database. The identification problem can be regarded as a largely solved problem, which even works in the presence of noise and slight temporal distortions of the query, see, e.g., [1, 2] and the references therein. Current identification systems, however, are not suitable for a less strict notion of similarity. The goal of *music synchronization* in the audio domain, sometimes also referred to as audio matching, is to time-align two given versions of the same underlying piece of music, see, e.g., [3, 4]. In contrast, the goal in our audio matching scenario is to identify short audio fragments similar to the query hidden in the database.

Finally, we mention two general approaches for feature design relevant to this paper. The *chroma-based approach* as suggested by [5] 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 signal, which is 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., [6]. In the next section, we will explain in detail how we combine both approaches by evaluating chroma-based audio features by means of short-time statistics.

2. AUDIO FEATURES

In the design of audio features one often has to deal with the problem of satisfying two conflicting goals at the same time: robustness to admissible variations on the one hand and accuracy with respect to the relevant characteristics on the other hand. Furthermore, the features should support an efficient algorithmic solution of the problem they are designed for. In our audio matching scenario, we consider audio clips as similar if they represent the same

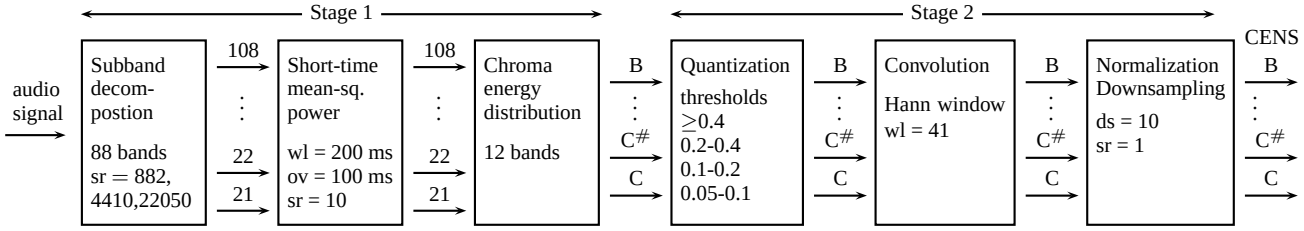

Figure 1: *Two-stage CENS feature design (wl = window length, ov = overlap, sr = sampling rate, ds = downsampling factor).*

musical content regardless of the specific interpretation and instrumentation. In other words, the audio matching procedure has to be invariant under parameters such as timbre, dynamics, articulation, and local tempo deviations as well as under slight variations in note groups such as trills or grace notes. Furthermore, global tempo and global pitch transpositions should be accounted for. It turns out that the rough harmonic progression over some period of time (10 to 30 seconds, as our experiments show) often characterizes a piece of music to a high degree, while still being invariant under the above parameters. This motivates the two-stage feature design as described in this section, see also Fig. 1.

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. 2.1). Using chroma distributions not only takes into account the close octave relationship in both melody and harmony as typical for Western music, see also [5], but also introduces a high degree of robustness to variations in dynamics, timbre, and articulation. In the second stage, we use a much larger statistics window to compute thresholded short-time statistics over these chroma energy distributions in order to introduce robustness to local time deviations and additional notes (Sect. 2.2). In the following, we identify the musical notes A0 to C8 (the range of a standard piano) with the MIDI pitches $p = 21$ to $p = 108$.

2.1. First stage: local chroma energy distribution features

First, we decompose the audio signal into 88 frequency bands with center frequencies corresponding to the MIDI pitches $p = 21$ to $p = 108$. To properly separate adjacent pitches, we need filters with narrow passbands, high rejection in the stopbands, and sharp cutoffs. In order to design a set of filters satisfying these stringent requirements for all MIDI notes in question, we work with three different sampling rates: 22050 Hz for high frequencies ($p = 96, \ldots, 108$), 4410 Hz for medium frequencies ($p = 60, \ldots, 95$), and 882 Hz for low frequencies ($p = 21, \ldots, 59$). Each of the 88 filters is realized as eighth-order elliptic filter with 1 dB passband ripple and 50 dB rejection in the stopband. To separate the notes we use a Q factor (ratio of center frequency to bandwidth) of $Q = 25$ and a transition band of half the width of the passband. To compensate the large phase distortions inherent to the elliptic filters, we use the standard technique of forward-backward filtering resulting in a zero phase distortion, see, e.g., [7].

Afterwards, we compute the short-time mean-square power (STMSP) for each of the 88 subbands by convolving the squared subband signals by a 200 ms rectangular window with an overlap of half the window size. Note that the actual window size depends on the respective sampling rate of 22050, 4410, and 882 Hz, which is compensated in the energy computation by introducing an additional factor of 1, 5, and 25, respectively. Then, we compute STMSPs of all chroma classes $\mathrm{C}, \mathrm{C}^{\#}, \ldots, \mathrm{B}$ by adding up the corresponding STMSPs of all pitches belonging to the respective class. For example, to compute the STMSP of the chroma A, we add up the STMSPs of the pitches A0,A1,. . .,A7. This yields for every 100 ms a real 12-dimensional vector $\vec{v} = (v_1, v_2 \ldots, v_{12}) \in \mathbb{R}^{12}$, v_1 corresponding to chroma C, v_2 to chroma $\mathrm{C}^{\#}$, and so on. Finally, we compute the energy distribution relative to the 12 chroma classes by replacing $\vec{v}$ by $\vec{v}/(\sum_{i=1}^{12} v_i)$.

In summary, in the first stage the audio signal is converted into a sequence $(\vec{v}^1, \vec{v}^2, \ldots, \vec{v}^N)$ of 12-dimensional chroma distribution vectors $\vec{v}^n \in [0,1]^{12}$ for $1 \leq n \leq N$. For the Vivaldi example given in the introduction, the resulting sequence is shown in Fig. 2 (light curve). Furthermore, to avoid random-like energy distributions occurring during passages of very 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 such passages. We also tested the short time Fourier transform (STFT) to compute the chroma features by pooling the spectral coefficients as suggested in [5]. Even though obtaining similar features, our filter bank approach, while having a comparable computational cost and allowing a better control over the frequency bands, produced slightly better results. This particularly holds for the low frequencies which is due to the more adequate resolution in time and frequency.

2.2. Second stage: normalized 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 further process the chroma features by taking a kind of thresholded short-time statistics. To quantize the chroma energy distribution vectors $\vec{v}^n = (v_1^n, \ldots, v_{12}^n) \in [0,1]^{12}$, we introduce a quantization function $Q : [0:1] \to \{0,1,2,3,4\}$ by letting

$$Q(a) := \begin{cases} 4 & \text{for} \quad 0.4 \leq a \leq 1, \\ 3 & \text{for} \quad 0.2 \leq a < 0.4, \\ 2 & \text{for} \quad 0.1 \leq a < 0.2, \\ 1 & \text{for} \quad 0.05 \leq a < 0.1, \\ 0 & \text{for} \quad 0 \leq a < 0.05. \end{cases}$$

Then, we define $Q(\vec{v}^n) := (Q(v_1^n), \ldots, Q(v_{12}^n))$ by applying Q to each component of $\vec{v}^n$. Intuitively, this quantization assigns the value 4 to a chroma component v_i^n if the corresponding chroma class contains more than 40 percent of the signal's total energy within the respective analysis window, and so on. The thresholds are chosen in a logarithmic fashion. Furthermore, chroma components below a 5 percent threshold are excluded from further considerations. In a subsequent step, we convolve the sequence $(Q(\vec{v}^1), \ldots, Q(\vec{v}^N))$ 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

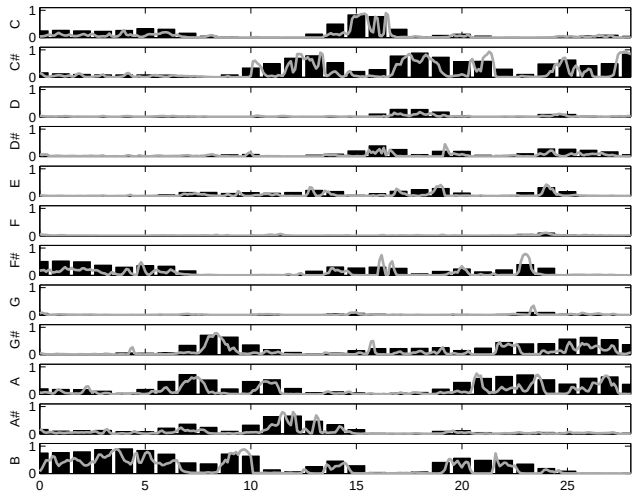

Figure 2: *Zukerman's interpretation of Vivaldi's Spring RV269, No. 1. The shown excerpt corresponds to measures 44–55, which correspond to* 28 *seconds in the recording (seconds 112–140). The light curves represent the local chroma energy distributions (10 features per second). The dark bars represent the CENS feature sequence (1 feature vector per second).*

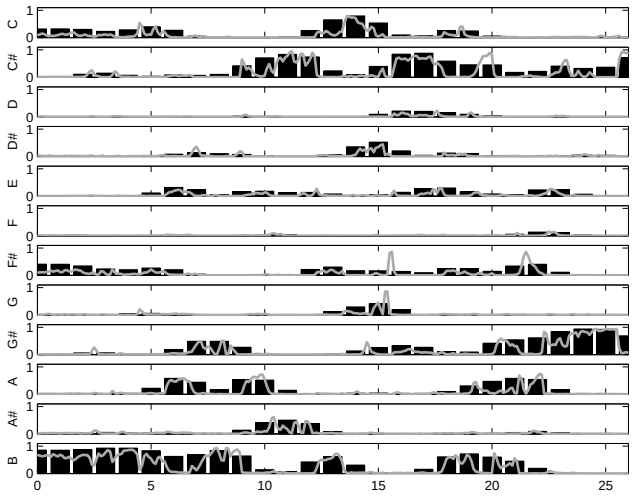

Figure 3: *CENS feature sequence for seconds 102-128 of Perlman's audio recording corresponding to measures 44–55 of the same Vivaldi example as in Fig. 2.*

statistics of the energy distribution over a window of 41 consecutive vectors. In a last step, this sequence is downsampled by a factor of 10. The resulting vectors are normalized with respect to the Euclidean norm. Altogether, one obtains one vector per second corresponding to roughly 4100 ms of audio. For short, these features vectors are simply referred to as *CENS* (**C**hroma **E**nergy distribution **N**ormalized **S**tatistics). Fig. 2 shows the resulting sequence of CENS feature vectors for our Vivaldi example.

In summary, the small analysis 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 (concerning the actual time span) statistics window. Note that simply enlarging the analysis window in the first stage omitting the second stage averages out valuable local harmonic information leading to less meaningful features. Taking local statistics in the second stage not only smooths out local time deviations as may occur for articulatory reasons but also compensates for different realizations of note groups such as trills or arpeggios. Here our two stage approach admits a high degree of flexibility in the feature design to find a good compromise between the two conflicting goals mentioned above.

2.3. Global tempo variations and global pitch transpositions

Various interpretations of the same underlying piece of music may differ considerably in the global tempo. For example, Mae's interpretation of Vivaldi's first movement of the Spring is 14 percent faster than the Zukerman interpretation. 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 the second stage. 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 $10/13 \approx 0.77$ relative to the original tempo. In our experiments, we used 8 different query versions corresponding to the downsampling factors $7, \ldots, 14$ covering global tempo variations of roughly -40 to $+40$ percent. Actually, one can employ a similar strategy to retrieve audio clips which are transposed versions of the query. The idea is to create different versions of the query by transposing it into all of the 12 existing keys. This is simulated by cyclically shifting the components of all CENS vectors extracted from the query and then again processing all these query versions independently.

3. AN APPLICATION TO AUDIO MATCHING

The CENS features were designed for robust and efficient audio matching and audio synchronization tasks. The goal of this section is to summarize the main ideas of our audio matching procedure and to report on some of our experiments illustrating the power of the CENS features. The details of our audio matching procedure will be published elsewhere. Further experimental material, visualizations as well as audio examples can be found at `www-mmdb.iai.uni-bonn.de/projects/audiomatching`.

Our test database contains 112 hours of audio material (mono, 22050 Hz) requiring 16.5 GB of disk space. It comprises 1167 audio files reflecting a wide range of classical music including pieces by Bach, Bartok, Bernstein, Beethoven, Chopin, Dvorak, Elgar, Mozart, Orff, Ravel, Schubert, Shostakovich, Vivaldi, and Wagner. The collection was set up to contain several different versions of most of the included pieces. Some of the orchestral pieces exists also as piano arrangements or synthesized MIDI-versions. In a preprocessing step, we compute the CENS feature sequences of all audio recordings contained in the database. By concatenating the individual sequences (keeping track of recording boundaries in a supplemental data structure) this results in a CENS feature sequence denoted as $\mathcal{D} := (\vec{v}^1, \vec{v}^2, \ldots, \vec{v}^N)$. Storing the features $\mathcal{D}$ 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 $\mathcal{D}$ is all we need during the matching procedure.

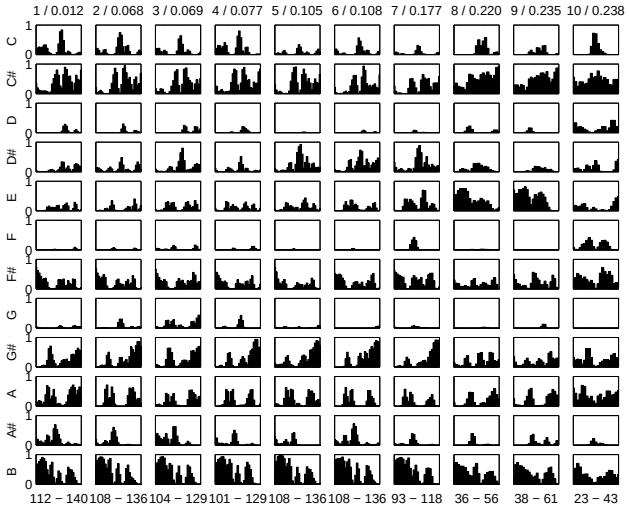

Figure 4: *The CENS feature sequences of the first ten matches for the Vivaldi query corresponding to Fig. 2.*

In our audio matching scenario, a typical query consists of a short audio clip of duration between 10 to 30 seconds. We first convert the query into a CENS feature sequence, in the following denoted by $\mathcal{Q} := (\vec{w}^1, \vec{w}^2, \ldots, \vec{w}^M)$. Then, we compare the sequence $\mathcal{Q}$ to any subsequence $(\vec{v}^i, \vec{v}^{i+1} \ldots, \vec{v}^{i+M-1})$ of $\mathcal{D}$ consisting of M consecutive vectors, where $1 \leq i \leq N - M + 1$. More specific, we define $\Delta(i) := 1 - \frac{1}{M}\sum_{m=1}^{M}\langle \vec{v}^{i+m-1}, \vec{w}^m \rangle$, taking the difference of one and the averaged inner product of corresponding CENS vectors in the sequences. Recall that the CENS vectors are normalized with respect to the Euclidean norm, i.e., the inner products $\langle \vec{v}^{i+m-1}, \vec{w}^m \rangle$ are equal to the cosine of the angle between $\vec{v}^{i+m-1}$ and $\vec{w}^m$. From this we obtain a distance function $\Delta : [1 : N - M + 1] \rightarrow [0, 1]$, where $\Delta(i)$ describes the distance of $\mathcal{Q}$ and the subsequence of $\mathcal{D}$ starting at position i and consisting of M consecutive vectors. To simulate global tempo variations as well as global pitch transformations, we produce additional query sequences as described in Sect. 2.3. For each of these sequences we compute a separate distance function, from which we then take at all time stamps the point-wise minimum. The resulting overall minimum distance function is again denoted by Δ.

To determine the best match between $\mathcal{Q}$ and $\mathcal{D}$, we simply look for the index $i_{\min} \in [1 : N - M + 1]$ minimizing Δ. Then the best match is given by the audio clip corresponding to the feature sequence $(\vec{v}_{i_{\min}}, \vec{v}_{i_{\min}+1} \ldots, \vec{v}_{i_{\min}+M-1})$. To look for the second best match, we exclude a neighborhood around the index $i_{\min}$ from further consideration to avoid "collisions" with the best match. To find subsequent matches, the latter procedure is repeated until a certain number of matches is obtained or a specified distance threshold is exceeded.

As example, we consider the query consisting of the Vivaldi audio clip shown in Fig. 2 (Zukermans's interpretation of Vivaldi's Spring RV269, No. 1, measures 44-55). Our database contains seven different interpretations of this piece (Abbado, Carmirelli, Lizzio, Mae, Nishizaki, Perlman, Zukerman). Using our matching procedure, we successively determined the best matches. As a remarkable result, the seven best matches exactly coincide with the audio excerpts in the seven interpretations corresponding to the measures of the query. Fig. 4 illustrates the CENS feature sequences of the best ten matches. 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 ($112 - 140$, 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 query has been cut from the original audio file resulting in a slight shift in the CENS features. The second best match has a Δ-distance of 0.068 and corresponds to seconds 108-136 of the Lizzio interpretation. Even the excerpt of the Mae interpretation, which significantly differs from the query in articulation and global tempo and which includes additional notes, was retrieved as seventh and last "correct" match with a Δ-distance of 0.177. The eighth best match, already having a Δ-distance of 0.220, is the first "false" match corresponding to some seemingly unrelated segment (seconds 36–56) of the Zukerman interpretation of the third movement of Vivaldi's "Spring". The 10th match even corresponds to some segment of Bach's Sinfonia No. 12, BWV798 for piano. All of these "false" matches, however, still reveal a harmonic progression similar to the query.

4. RESULTS, CONCLUSIONS, FUTURE WORK

Further matching results of our extensive experiments are available at `www-mmdb.iai.uni-bonn.de/projects/audiomatching`. As it turns out, our audio matching procedure performs well for most of query examples within a wide range of classical music proving the usefulness of our CESN features. The top matches almost always included the "correct" ones, even in case of synthesized MIDI versions and interpretations in other instrumentations. Among the top matches, however, there generally is also a small number of "false" matches, which may differ considerably from the query (accidentally having a similar harmonic progression). Here, one has to revert to other methods to automatically separate the "correct" from the "false" matches. This kind of postprocessing, however, can then be done on a significantly reduced data set.

5. REFERENCES

[1] 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.

[2] A. Wang, "An Industrial Strength Audio Search Algorithm," in *Proc. ISMIR, Baltimore, USA*, 2003.

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

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

[5] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. on Multimedia*, vol. 7, no. 1, pp. 96–104, Feb. 2005.

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

[7] M. D. Proakis, J.G., *Digital Signal Processsing*. Prentice Hall, 1996.