

Beat-ID: Identifying Music via Beat Analysis

Darko Kirovski and Hagai Attias

Microsoft Research

One Microsoft Way, Redmond, WA 98052, USA

Email: {darkok,hagaia}@microsoft.com

Abstract—Music identification is an effective tool that enables multimedia players to extract a distinct statistical digest of the played content, look up into a music database using the extracted unique identifier, and then take advantage of the services available for that particular content. In this paper, we introduce Beat-IDs, the first music identification system that creates the digest of the music clip by understanding the basic structure of every musical piece: its beat. A Beat-ID is created in two steps: first, the system detects the average beat period of a given music clip using a modified EM algorithm and then, it analyzes the statistical properties of the clip with respect to the detected beats. The extracted 32-byte Beat-ID contains two components: the length of the average beat period and a compressed statistical digest of signal's energy distribution in an average beat period. Finally, we introduce an algorithm for matching Beat-IDs that quantifies the matching accuracy between two music identifiers using an error analysis. In this paper, the properties of Beat-IDs are demonstrated using a relatively small database of audio clips.

I. INTRODUCTION

WITH the introduction of compact music formats such as MP3, Real Audio, and Windows Media Audio, music downloading has become one of the most popular activities on the Internet. Simultaneously, numerous Web services have been established to accompany music downloading from several perspectives including e-commerce, Web-search, recommendation systems, and broadcast monitoring. One of the key technologies that enables content-targeted services is music identification¹. The main goal of such a technology is to compute a distinct identifier for a given clip which is at maximal distance with respect to the identifiers of all other music clips in a given large database. The assumption is that music clips have distinct features such as a unique melody, rhythm, and playing orchestra. Based on this assumption, we expect that an identifier of a musical piece that extracts this information should uniquely and accurately vouch for clip's contents. Clearly, if two clips have similar contents (e.g. a studio and live recording of the same song), we cannot expect that their identifiers are largely different.

The definition of a music identifier points to a natural and highly accurate implementation: extraction of music notes – melody and rhythm – and recognition of the playing orchestra and vocals. Since analyzing such information in a given audio clip is hard and requires significant computing resources, in this paper, we take the first step towards such a system. We build

music identifiers, Beat-IDs, by analyzing the fundament of every musical piece: its beat². We analyze two features of music with respect to its beat: the average beat period and the average pattern of music events across all detected beats. Although different musical pieces may have similar melodies, the pattern of musical events (e.g. the tempo of striking notes on an instrument) in two musical pieces is unlikely to be similar. Clearly, such an analysis does not capture all features of a musical piece – mainly the pitch of the played notes. We estimate that the system can be significantly improved by concatenating digests of uncovered music features. Nevertheless, in this paper, we focus on beat analysis and demonstrate the accuracy that can be achieved in identifying music content using Beat-IDs.

A. Music Identification via Beat-IDs

The Beat-ID system consists of several main components. The first component is a modified estimation maximization (EM) algorithm [1] that aims at accurately estimating beat periods within a given clip. As a side result, the beat detector returns clip's average beat period. The second component statistically analyzes the energy distribution in the time domain across all detected beats. As a result, a Beat-ID, a compact 32-byte representation of a music clip, consists of the following information:

- Length of an average beat period, and
- Compact representation of the shape of the averaged energy distribution in the time domain across all detected beat periods.

A music identification repository is a sorted table of entries, where each entry corresponds to a particular song and contains its Beat-ID and the variance for each individual feature of that Beat-ID. For example, the length of the average beat period is accompanied with the variance among individual beat periods within a given clip. Since a Beat-ID is created as a collection of averages of various statistical features of the audio clip, the purpose of storing the variances of corresponding features is to quantify the accuracy of a match-up between two Beat-IDs. A

²Music is rigorously defined using a strict notation as follows. The pulse, or pattern of regular accents, of a musical piece can be broken into individual pulses, or beats. In rhythmic notation, notes are assigned time values by their relation to these beats. The grouping of beats in a piece of music establishes the music's meter. Meter is identified by the time signature, a fractional symbol in which the numerator specifies the number of beats per bar, and the denominator specifies the relative note value assigned to one beat. A time signature of 4/4 indicates four beats per measure and the fourth note is given a value of one beat.

¹Also referred to as music fingerprinting or hashing.

lookup into a database is performed only using the Beat-ID of the unknown musical piece.

In order to facilitate the search process, we sort the table of Beat-IDs using a sequence of prioritized criteria, where each criterion corresponds to an ascending or descending order of a particular individual feature of a Beat-ID. The table can be additionally indexed using standard techniques well-researched in the database systems world. The best matched song entry upon a lookup into the Beat-ID database, is associated with an accuracy quantifier – the likelihood of an accurate matching computed using a statistical analysis that takes into account the recorded variances for all Beat-ID features.

B. Related Work

Multimedia identification is a relatively new field with efforts targeting audio, images, and video. Burges et al. introduce distortion discriminant analysis as a technique for extracting the perceptual uniqueness from an audio clip [2]. Kalker et al. present an effective energy thresholding technique for computing robust audio hashes [3]. Mihcak and Venkatesan rely on error-correcting codes and statistical analysis of randomly sized and located polygons in image and time-frequency audio planes to create cryptographically secure multimedia content hashes [4]. In this paper, we demonstrate the first technology for music identification that uses detection of *semantic* music events to establish perceptual song digests, rather than a "blind" statistical analysis of the multimedia content that dominates previous works.

II. BEAT DETECTION

One of the most robust events in music is its beat. An overwhelming portion of popular and in particular, classical music is defined using a well-known precise notation that rigorously dictates the playing tempo. Rarely, music content experiences intentional and significant variance in rhythm and even when it happens it does not last more than several seconds. In summary, although music is authored in diverse ways, one characteristic that is predominant for music is the rigorous constraint for accurate rhythm. We use this fact to create an algorithm for beat detection as a principal component of the Beat-ID system.

We have developed an EM-based algorithm [1], [6] for beat detection that consists of three steps. First, the mean period of the beat is estimated from data using a statistical modeling approach. Second, the mean onset of the beat is estimated. Third, the actual onset of the beat for each beat period is estimated. An important comment is that we have applied this algorithm to 12-second music windows with non-significant rhythm changes. Other cases can be trivially reduced to the problem being solved here using traditional segmentation algorithms.

Mean beat period. Let u_m denote the signal energy at frame m . To compute u_m , we consider the signal waveform in the time domain, and apply a window function at equally spaced time points indexed by $m = 1, \dots, M$. u_m is the mean squared value of the windowed signal.

We model the beat by assuming that u_m is approximately periodic in m , with beat period τ . To estimate τ we use the following model,

$$u_m = au_{m-\tau} + v_m, \quad (1)$$

where v_m is i.i.d. Gaussian noise with mean zero and variance σ^2 . Hence we have a probabilistic model where u_m are the observed variables, τ is a hidden variable, and a, σ are parameters:

$$p(\{u_m\} | \tau) = \prod_m \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(u_m - au_{m-\tau})^2 / 2\sigma^2}. \quad (2)$$

To complete the definition of our model, we must specify a prior distribution $p(\tau)$ over the beat period. We use a flat distribution $p(\tau) = \text{const.}$.

We now use an EM algorithm to estimate the period τ and the model parameters. As usual with EM, this is an iterative algorithm, where the E-step updates the sufficient statistics, and the M-step updates the parameter estimates. In our case, the sufficient statistics is the full posterior distribution over the beat period conditioned on the data. It is computed via Bayes' rule,

$$p(\tau | \{u_m\}) = \frac{1}{z} p(\{u_m\} | \tau) p(\tau), \quad (3)$$

where z is a normalization constant. It can be shown to equal the data distribution, $z = p(\{u_m\})$, but since it is independent of τ it does not need to be actually computed. This posterior can be computed efficiently for any value of τ by observing that its logarithm is the autocorrelation of u_m ,

$$\log p(\tau | \{u_m\}) = \frac{1}{\sigma^2} \sum_m u_m u_{m-\tau} + \text{const.} \quad (4)$$

and using FFT. The resulting complexity of the E-step is $\mathcal{O}(M \log M)$.

The M-step update rules are derived by minimizing the complete data log-likelihood $E \log p(\{u_m\} | \tau) p(\tau)$, where the operator E performs averaging over τ w.r.t. the posterior (3). We obtain

$$\begin{aligned} a &= \sum_m u_m E u_{m-\tau} / \sum_m u_m^2 \\ \sigma^2 &= \frac{1}{M} \sum_m E (u_m - a u_{m-\tau})^2. \end{aligned} \quad (5)$$

As in the E-step, the computations involved in (5) can be performed efficiently using FFT.

Finally, the beat period is obtained using a MAP estimate,

$$\hat{\tau} = \arg \max_{\tau} p(\tau | \{u_m\}). \quad (6)$$

Experimentally, the posterior over τ turns out to be quite narrow. Below we use τ to refer to $\hat{\tau}$.

Mean beat onset. To compute this quantity, we divide u_m into consecutive non-overlapping sequences of length τ . Denote sequence i by (u_1^i, \dots, u_τ^i) , where $u_n^i = u_{(i-1)\tau+n}$ and

$n = 1, \dots, \tau$. We then average over those sequences. Denote the average sequence by $(\bar{u}_1, \dots, \bar{u}_\tau)$, then the mean onset \bar{l} is its maximum,

$$\bar{l} = \arg \max_{1 \leq n \leq \tau} \bar{u}_n. \quad (7)$$

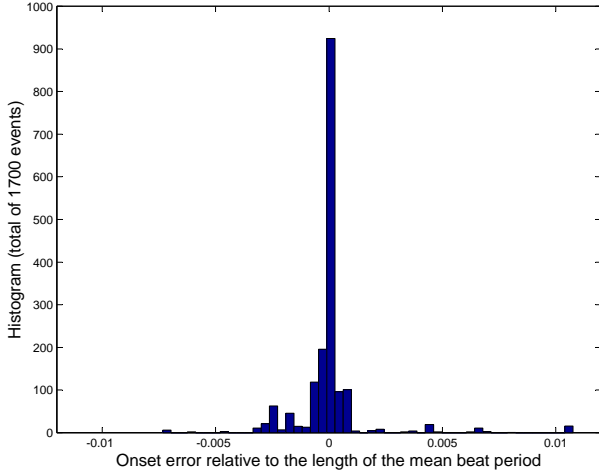


Fig. 1. Histogram of the error in the location of an actual beat onset with respect to the beat period in a typical pop-song mixed with an additive white gaussian noise conceived as $\mathcal{N}(0, 0.1)$. More than 99% of beats were detected with sub-percent relative accuracy.

Actual beat onset. This quantity is computed for each of the τ -long sequences above. We assume that the onset time l for i given sequence may deviate from the mean onset \bar{l} by as much as 10% of the beat period. Hence, we search for l_i , the beat onset time for sequence i , within the corresponding interval,

$$l_i = \arg \max_{\bar{l} - \tau/10 \leq n \leq \bar{l} + \tau/10} u_n^i. \quad (8)$$

The onset times l_i are converted back to the time domain and form the output beat signal. The most important characteristic of this beat detector is robustness of its decisions in the presence of strong noise. In an experiment on 50 different audio clips that ranged from classical to pop, we compared the actual beat onsets detected in the original clips to the actual beat onsets detected in the same set of clips with added zero-mean i.i.d. gaussian noise with standard deviation equal to one tenth of the maximum signal amplitude. Large percentage of over 99% of actual beat onsets was detected in neighborhoods of their original detected locations within 1% of the length of the mean beat period. A short summary of this experiment is illustrated in Figure 1.

III. COMPUTING THE BEAT-ID

The computation of a Beat-ID is primarily based on the detected beat patterns. There are three preprocessing steps: (i) signal normalization to prevent sensitivity to different attenuation levels and (ii) resampling down to 16kHz and (iii) psycho-acoustic filtering [5], which preserve the most significant portion of the music signal while reducing dependency with respect

to strong high-frequency noise and/or low bitrate compression codecs. As a main processing step, the beat detector computes the following information:

- τ – the length of the mean beat period and
- v – the energy of the signal averaged across beat periods:

$$v = \frac{1}{M} \sum_{i=1}^M u_{l_i}^i \quad (9)$$

where $u_{l_i}^i$ is defined as in Eqn.8.

This information represents the basis for computing the Beat-ID. The shape of v relatively accurately captures the rhythmic events in music, while ignoring the information about the pitch. The expectation is that musical pieces rarely follow the same rhythmic pattern while applying a different melody. In that sense, this statistic along with the detected tempo τ is sufficient to discriminate music clips.

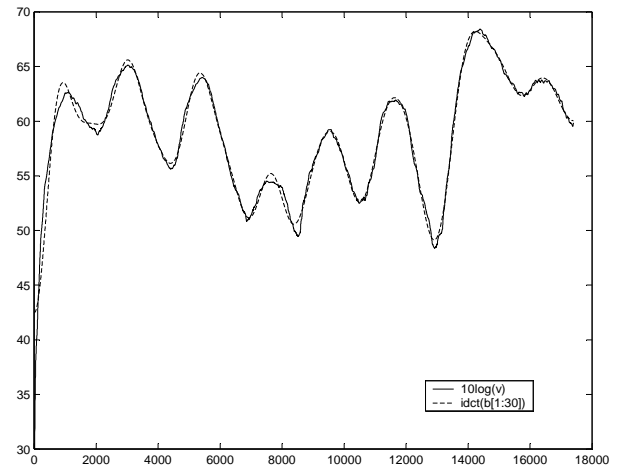


Fig. 2. An example of the signal $b = 10 \log(v)$ that represents the averaged energy across all beats in a 12 second samba clip and the reversed signal represented by low-pass filtering of b . Only the first 30 DCT subbands are preserved.

It is desirable to represent the unique identifier with as few bits as possible for two reasons: size of the Beat-ID repository and communication payload for a database lookup. Hence, we compress the beat-averaged energy spectrum via a DCT filter applied onto the dB-magnitude of the signal v :

$$b = DCT(10 \log(v)). \quad (10)$$

Since the high-frequency noise in v rarely represents the music features of a song, we apply a low-pass filter onto b to preserve the smooth shape of the energy distribution signal v . Typically, we preserve the first 30 sub-bands of b and weight-quantize them with 8 bits each. The weighting normalizes the variance for each individual feature within a large Beat-ID repository. Once this repository is assembled, the weight factors are computed for each Beat-ID feature and distributed to Beat-ID clients. A realistic assumption is that additional updates to a large Beat-ID repository are not likely to significantly change the optimal weight factors.

A total byte-length of a Beat-ID equals 32 bytes: 30 bytes for encoding $b[1] \dots b[30]$ and 2 bytes for encoding τ . Since a Beat-ID is computed over a period of 10-20 seconds of audio with relatively constant rhythm, typically a single song may have in its entry in the Beat-ID repository more than one Beat-ID. For most applications, two Beat-IDs are sufficient; the first one covers the beginning of the clip, while the second one covers the main theme of the clip.

IV. SEARCHING A BEAT-ID REPOSITORY

A lookup into the Beat-ID repository initially finds a small set of clips with Beat-ID feature values within certain relatively small distance with respect to the corresponding feature values of the unknown lookup Beat-ID. This process is performed using standard database search procedures, which are outside the scope of this paper. Each entry from the set of suspect clips is compared against the lookup Beat-ID. The procedure for comparing a *lookup* against a *suspect* Beat-ID is based on their mutual similarity and the variances of the suspect Beat-ID. The variances of the lookup Beat-ID are not considered in the matching process as they can be easily manipulated during sound editing. The matching confidence quantifier equals the probability that the features of the lookup Beat-ID x are that of the suspect Beat-ID y based on the variances recorded for this ID:

$$p(x \equiv y) = Q_\tau(\tau_x - \tau_y|y) \prod_{i=1}^{30} Q_{b[i]}(b_x[i] - b_y[i]|y), \quad (11)$$

where $Q_a(a_x - a_y|y)$ returns the tail probability that the lookup clip x that corresponds to the clip y has a Beat-ID feature value a_x at $|a_x - a_y|$ distance from a_y . The pdf of the variable $|a_x - a_y|$ is recorded empirically while generating the entry for clip y in the Beat-ID repository. The pdf is extrapolated based on the change of $|a_x - a_y|$ with respect to several instances of distinct noise additions. Since the pdf of $a_x - a_y$ is typically a zero-mean gaussian with standard deviation σ_a , then:

$$Q_a(a_x - a_y|y) = \text{erfc} \left(\frac{|a_x - a_y|}{\sigma_a \sqrt{2}} \right). \quad (12)$$

A lookup x into the Beat-ID repository returns the index of the best matched entry y as well as the match separation ratio of $\delta(x) = \log[p(x \equiv y)/p(x \equiv z)]$, where z is the second best matched clip from the Beat-ID repository.

V. EXPERIMENTAL RESULTS

One of the strong benefits of Beat-IDs is that the basic step of the sliding search-window is much longer than in "blind" identification systems. The basic step can be as low as one beat, but typically steps of 4-5 beats (>2sec) yield accurate detection. This has two-fold advantage: (a) search is significantly faster and (b) there are fewer tests, so, at equal false alarm rates, Beat-IDs would have significantly fewer errors per minute of music.

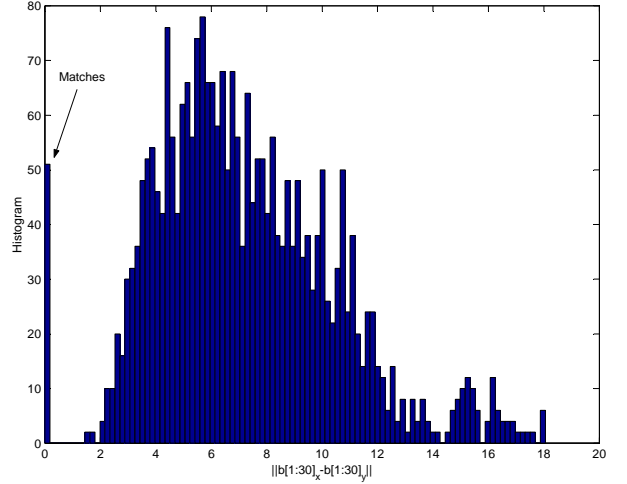


Fig. 3. An example of the separation of b -features for Beat-IDs from 51 distinct music clips. The histogram depicts the norm: $\sqrt{\frac{1}{30} \sum_{i=1}^{30} (b_x[i] - b_y[i])^2}$.

In our implementation, a real-time Beat-ID computation under these circumstances required about 15 MIPS.

The error rate of the Beat-ID depends on the lengths of average beat periods and the similarity of energy distribution patterns across beats among a set of clips. In our database of 51 clips, only one feature, τ , was sufficient to discriminate all but 4 clips with likelihood better than 10^{-3} . For this feature, the average minimal $Q_\tau(\tau_x - \tau_y|y) < 10^{-4}$, where the pdfs on τ for each clip were collected as presented in Figure 1. To analyze the discrimination achieved with vector $b[1 : 30]$, in the experiment, we have assumed that all songs in the database had the same τ . The match separation ratios (with assumed equivalent τ features) were $\delta(x) > 4$ for all clips x in the database. Hence, we concluded that for a detection threshold $\delta(x) \leq T = 4$, assuming a false negative, the estimated worst-case false positive rate was better than 10^{-6} .

The presented results may change when a music library contains clips with similar rhythmic content. Although we could not find such music clips, in the case when there exists such a library, we propose concatenating Beat-IDs with other forms of music identifiers which capture different music features such as pitch or vocals.

REFERENCES

- [1] Dempster A.P., Laird N.M., Rubin D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, **39**(1):1–38, (1977).
- [2] Burges C.J.C., Platt J.C., Jana S.: Extracting Noise-Robust Features from Audio Data. *ICASSP*, 2002.
- [3] Haitsma J.A., Kalker T., Oostveen J.: Robust Audio Hashing for Content Identification. *Content Based Multimedia and Indexing*, 2001.
- [4] Mihcak M.K., Venkatesan R.: RobustAudio Hashing. *Info Hiding Workshop*, 2001.
- [5] Malvar H.S.: Auditory Masking in Audio Compression. *Audio Anecdotes*, 2000.
- [6] Frey B., Jojic N.: Fast, Large-Scale Transformation-Invariant Clustering. *NIPS*, 2001.