Real-time beat estimation using feature extraction

Kristoffer Jensen and Tue Haste Andersen

Department of Computer Science, University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen, Denmark,
{krist,haste}@diku.dk,
WWW home page: http://www.diku.dk/~krist

Abstract. This paper presents a novel method for the estimation of beat interval from audio files. As a first step, a feature extracted from the waveform is used to identify note onsets. The estimated note onsets are used as input to a beat induction algorithm, where the most probable beat interval is found. Several enhancements over existing beat estimation systems are proposed in this work, including methods for identifying the optimum audio feature and a novel weighting system in the beat induction algorithm. The resulting system works in real-time, and is shown to work well for a wide variety of contemporary and popular rhythmic music. Several real-time music control systems have been made using the presented beat estimation method.

1 Introduction

Beat estimation is the process of predicting the musical beat from a representation of music, symbolic or acoustic. The beat is assumed to represent what humans perceive as a binary regular pulse underlying the music. In western music the rhythm is divided into measures, e.g. pop music often has four beats per measure. The problem of automatically finding the rhythm include finding the time between beats (tempo), finding the time between measures, and finding the phase of beats and measures. This work develops a system to find the time between beats from a sampled waveform in real-time. The approach adopted here consists of identifying promising audio features, and subsequently evaluating the quality of the features using error measures.

The beat in music is often marked by transient sounds, e.g. note onsets of drums or other instruments. Some onset positions may correspond to the position of a beat, while other onsets fall off beat. By detecting the onsets in the acoustic signal, and using this as input to a beat induction model, it is possible to estimate the beat.

Goto and Muraoka [1] presented a beat tracking system, where two features were extracted from the audio based on the frequency band of the snare and bass drum. The features were matched against pre-stored drum patterns and resulted in a very robust system, but only applicable to a specific musical style.

Later Goto and Muraoka [2] developed a system to perform beat tracking independent of drum sounds, based on detection of chord changes. This system was not dependent on the drum sounds, but again limited to simple rhythmic structures. Scheirer [3] took another approach, by using a non-linear operation of the estimated energy of six bandpass filters as feature extraction. The result was combined in a discrete frequency analysis to find the underlying beat. The system worked well for a number of rhythms but made errors that related to a lack of high-level understanding of the music. As opposed to the approaches described so far Dixon [4] built a non-causal system, where an amplitude-based feature was used as clustering of inter-onset intervals. By evaluating the inter-onset intervals, hypotheses are formed and one is selected as the beat interval. This system also gives successful results on simpler musical structures.

The first step of this work consists of selecting an optimal feature. There are a very large number of possible features to use in segmentation and beat estimation. Many audio features are found to be appropriate in rhythm detection systems, and one is found to perform significantly better. The second step involves the introduction of a high-level model for beat induction from the extracted audio feature. The beat induction is done using a running memory module, the beat probability vector, which has been inspired by the work of Desain [5].

The estimation of beat interval is a first step in the temporal music understanding. It can be used in extraction and processing of music or in control of music. The beat detection method presented here is in principle robust across music styles. One of the uses of the beat estimation is in beat matching, often performed by DJs using contemporary electronic and pop music. For this reason, these music styles has mainly been used in the evaluation. The system is implemented in the open source DJ software Mixxx [6] and has been demonstrated together with a baton tracking visual system for the use of live conducting of audio playback [7].

2 Audio Features

The basis of the beat estimation is an audio feature that responds to the transient note onsets. Many features have been introduced in research of audio segmentation and beat estimation. Most features used here have been recognized to be perceptually important in timbre research [8]. The features considered in this work are: amplitude, spectral centroid, high frequency energy, high frequency content, spectral irregularity, spectral flux and running entropy, all of which have been found in the literature, apart from the high frequency energy and the running entropy.

Other features, such as the vector-based bandpass filter envelopes [3], or mel-cepstrum coefficients have not been evaluated. Vector-based features need to be combined into one measure to perform optimally, which is a non-trivial task. This can be done using for instance artificial neural nets [9] that demands a large database for training, or by summation [3] when the vector set is homogeneous.

Most features indicate the onsets of notes. There is, however, still noise on many of the features, and the note onsets are not always present in all features. A method to evaluate and compare the features is presented in section 3, and used in the the selection of the optimal feature. In the following paragraphs, a number of features are reviewed and a peak detection algorithm is described.

2.1 Features

The features are all, except the running entropy, computed on a short time Fourier transform with a sliding Kaiser window. The magnitude $a_{n,k}$ of block n and FFT index k is used. All the features are calculated with a given block and step size (N_b and N_s respectively).

The audio features can be divided into absolute features that react to specific information weighted with the absolute level of the audio and relative features that only react to specific information. The relative features are more liable to give false detection in weak parts of the audio.

The amplitude has been found to be the only feature necessary in the tracking of piano music [10]. This feature is probably useful for percussive instruments, such as the piano or guitar. However, the amplitude feature is often very noisy for other instruments and for complex music.

Fundamental frequency is currently too difficult to use in complex music, since it is dependent on the estimation method. It has been used [9] in segmentation of monophonic audio with good results, though.

One of the most important timbre parameters is the spectral centroid (brightness) [11], defined as:

$$SC_n = \frac{\sum_{k=1}^{N_b/2} k a_{n,k}}{\sum_{k=1}^{N_b/2} a_{n,k}}. \tag{1}$$

The spectral centroid is a measure of the relative energy between the low and high frequencies. Therefore it seems appropriate in the detection of transients, which contain relatively much high frequency energy.

An absolute measure of the energy in the high frequencies (HFE) is defined as the sum of the spectral magnitude above 4kHz,

$$HFE_n = \Sigma_{k=f_{4k}}^{N_b/2} a_{n,k}. \tag{2}$$

where f_{4k} is the index corresponding to 4 kHz.

Another absolute measure, the high frequency content (HFC) [12] is calculated as the sum of the amplitudes and weighted by the frequency squared,

$$HFC_n = \Sigma_{k=1}^{N_b/2} k^2 a_{n,k}. \tag{3}$$

These features are interesting because they indicate both high energy, but also relatively much high frequency energy.

The spectral irregularity (SPI), calculated as the sum of differences of spectral magnitude in one block,

$$SPI_n = \Sigma_{k=2}^{N_b/2} \left| a_{n,k} - a_{n,k-1} \right|, \tag{4}$$

and the spectral flux (SPF), calculated as the sum of spectral magnitude differences between two adjoining blocks,

$$SPF_n = \Sigma_{k=1}^{N_b/2} \left| a_{n,k} - a_{n-1,k} \right|, \tag{5}$$

are two features known from the timbre perception research. These features give indication of the noise level and the transient behavior that are often indicators of beats.

Note onsets can be considered as new information in the audio file. Therefore the running entropy, calculated on a running histogram of the 2^{16} quantization steps is considered. First the probability of each sample value is estimated for one block,

$$H_n(s(l)) = H_n(s(l)) + \frac{1}{N_b}, l = (n-1)N_s + 1 \cdots (n-1)N_s + N_b, \tag{6}$$

then the probability is updated with $1 - W_h$,

$$H_n = W_h H_n + (1 - W_h) H_{n-1}, \tag{7}$$

and finally the entropy in bits is calculated,

$$Ent_n = -\Sigma_{k=1}^{2^{16}} H_n(k) \log_2(H_n(k)). \tag{8}$$

These are the features evaluated in this work. The note-onsets are considered to occur at the start of the attacks, but the features generally peak at the end of the attacks. To compensate for this delay the time derivative is taken on the features. The second derivative is taken on the running entropy. The maximum of the derivative of the amplitude has been shown to be important in the perception of the attack [13]. In addition, the negative values of each feature are set to zero.

An example of the resulting time-varying extracted features can be seen in fig. 1 for a contemporary music piece[1]. On the figure manually marked note onsets are indicated by dashed lines. It is clear that most features peak at the note onsets. There is, however, still noise on many of the features, and some of the note onsets are not always present in the features.

2.2 Peak detection

The features considered in the previous section all exhibit local maximums at most of the perceptual note onsets. To identify a note onset from a given feature a peak detection algorithm is needed. The peak detection algorithm used here chooses all local maximums, potentially using a threshold,

$$p = (F_{n-1} < F_n > F_{n+1}) \wedge (F_n \geq th) \tag{9}$$

where F is an arbitrary audio feature. In addition to the peak detection, a corresponding weight, w_k is also calculated at each peak k (at the time t_k),

[1] Psychodelik. Appearing on LFO - Advance (Warp 039), January 1996.

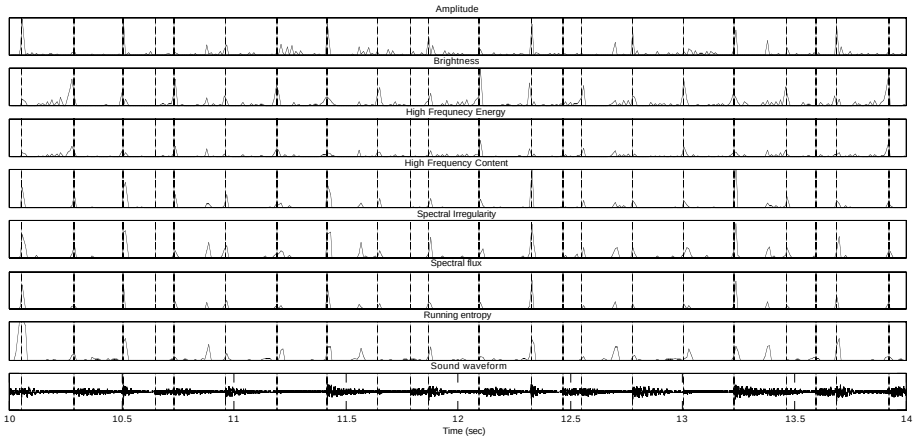

Fig. 1. Audio features from the LFO -Psychodelik piece (excerpt) as function of time. The features are shown at arbitrary scales. The vertical dashed lines indicate the manual marked transients.

corresponding to the time steps where p is true. This weight is later used in the beat probability vector, and in the detection of the phase of the beat. The threshold is used in the selection of the optimal feature, but not in the final beat estimation system.

3 Feature analysis

To compare features, different musical pieces has been analyzed manually by placing marks at every perceptual note onset. The marking consists in identifying the note onsets that are perceptually important for the rhythmic structure. These note onsets are generally generated by the hi-hat and bass drum and any instrument with a transient attack. In practice, some parts of the pieces lack hi-hat and bass drum, and the rhythmic structure is given by other instruments. The manual marking of the note onsets in time has an error estimated to be below 10 msec. In all eight musical pieces were used, with an average of 1500 note onsets per piece.

These manual marks are used as basis for comparing the performance of the various features. In order to select the optimum feature, three different error measures are used, based on matched peaks, that is peaks located within a time threshold (20 msec) to a manual mark. An unmatched peak is located outside the time threshold from a manual mark.

3.1 Error measures

To find the signal to noise the value of a matched (P), or unmatched ($\hat{P}$) peak is calculated as the sum of the feature at both sides of the peak where the slope

is continually descending from the peak center. The signal to noise ratio is then calculated as,

$$s_n = \frac{\Sigma_{n=0}^{N_{matched}} P_n}{\Sigma_{n=0}^{N_{unmatched}} \hat{P}_n}. \tag{10}$$

The missed ratio is calculated as the number of manual marks minus the number of matched peaks, divided by the number of manual marks,

$$R_{missed} = \frac{N_{marked} - N_{matched}}{N_{marked}}, \tag{11}$$

and the spurious ratio is calculated as the number of unmatched peaks, divided by the number of manual marks,

$$R_{spurious} = \frac{N_{unmatched}}{N_{marked}}. \tag{12}$$

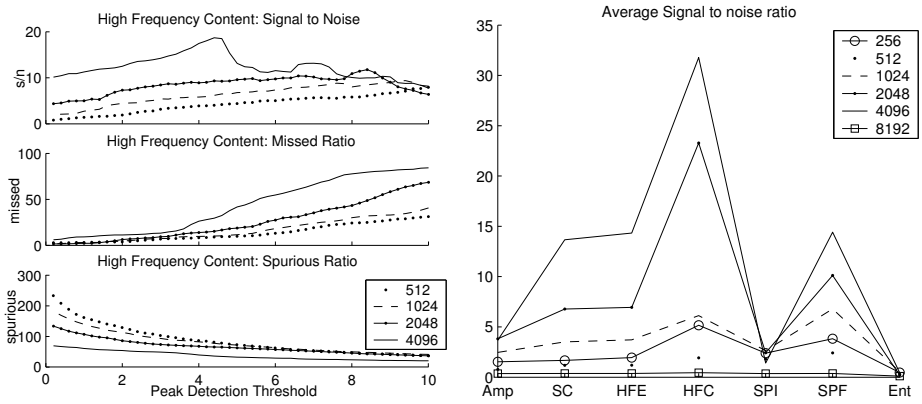

Fig. 2. Left: Example of error measures calculated using different block sizes of the HFC feature for the piece Train to Barcelona. Right: Average signal to noise for fixed threshold, and all music pieces for many block sizes and features.

3.2 Analysis and selection of feature

In order to evaluate the features the error measures are now calculated on the music material using a varying peak detection threshold. An example of the error measures for the piece Train to Barcelona[2] is shown in the left part of fig. 2. For low thresholds, there are few missed beats, and for high peak detection threshold, there are many missed beats. The spurious beats (false indications)

[2] By Akufen. Appearing on Various - Elektronische Musik - Interkontinental (Traum CD07), December 2001.

behave in the opposite way, for low thresholds there is up to several hundred percents, whereas the spurious ratio is low for high peak detection thresholds. Under these conditions it is difficult to select an optimum peak detection threshold, since both low missed and spurious ratio is the optimization goal and they are mutually exclusive. The signal to noise ratio generally rises with the peak detection threshold, which indicates that the few found peaks contain most of the energy for the high thresholds. There seem to be no optimum way of selecting the threshold.

An analysis of the error values for all features and music pieces gives no clear indication of the best feature. Therefore a different approach has been used.

Initial tests have shown that the beat estimation method presented in the next section need at least 75% of the note onsets to perform well. The threshold for 75% matched beats (25% missed) is therefore found for each features/block size pair and music piece. The signal to noise ratio is then found for this threshold. The average signal to noise ratio is calculated for all music pieces. The result is shown in the right part of fig. 2.

Several results can be obtained from the figure. First, it is clear that the extreme block sizes, 256, 512, and 8192 all perform inadequately. Secondly, several features also perform poorly, in particular the amplitude, the spectral irregularity, and the entropy. The best features are the spectral centroid, the high frequency energy, the high frequency content and the spectral flux. The HFC performs significantly better than the other features, in particular for the block sizes 2048 and 4096, which has the best overall signal to noise ratio.

4 Beat estimation

The analysis of the audio features has permitted the choice of feature and feature parameters. There is, however, still errors in the detected peaks of the chosen features. As described in other beat estimation systems found in the literature, a beat induction system, that is a method for cleaning up spurious beats and introducing missing beats, is needed. This could be, for instance, based on artificial neural nets, as in [9], but this method demands manual marking of a large database, potentially for each music style. Another alternative is the use of frequency analysis on the features, as in [3], but this system reacts poorly to tempo changes.

Some of the demands of a beat estimation system are stability and robustness. Stability to ensure that the estimation is yielding low errors for music exhibiting stationary beats and robustness to ensure that the estimation continues to give good results for music breaks without stationary beats. In addition, the system should be causal, and instantaneous. Causal to ensure real-time behavior, and instantaneous to ensure fast response.

These demands are fulfilled by the use of a memory-based beat probability vector that is based on the model of rhythm perception by Desain [5]. In addition a tempo range is needed to avoid the selection of beat intervals that do not occur

in the music style. The tempo is chosen in this work to lie between 50 and 200 BPM, which is similar to the constraints used in [3].

4.1 Beat probability vector

The beat probability vector is a dynamic model of the beat intervals that permits the identification of the beat intervals from noisy features. The probability vector is a histogram of note onset intervals, as measured from the previous note onset. For each new note onset the probability vector $H(t)$ is updated (along with its neighboring positions) by a Gaussian shape at the intervals corresponding to the distance to the previous peak. To maintain a dynamic behavior, the probability vector is scaled down at each time step. At every found peak k the peak probability vector is updated,

$$H(t) = W^{t_k - t_{k-1}} H(t) + G(t_k - t_{k-1}, t), t = 0 \ldots \infty \tag{13}$$

where W is the time weight that scale down the probability of the older intervals, and G is a Gaussian shape which is non-zero at a limited range centered around $t_k - t_{k-1}$. The current beat interval is identified as the index corresponding to the maximum in the beat probability vector, or, alternatively, to $t_k - t_{k-1}$ if the interval is located at the vicinity of the maximum in the beat probability vector.

The memory of the beat probability vector allows the detection of the beat interval in breaks with missing or alternative rhythmic structure. An instantaneous reaction to small tempo changes is obtained if the current beat interval is set to the distance between peaks at proximity to the maximum in the vector.

In [5] multiples of the intervals are also increased. Since the intervals are found from the audio file in this work, the erroneous intervals are generally not multiples of the beat. Another method must therefore be used to identify the important beat interval.

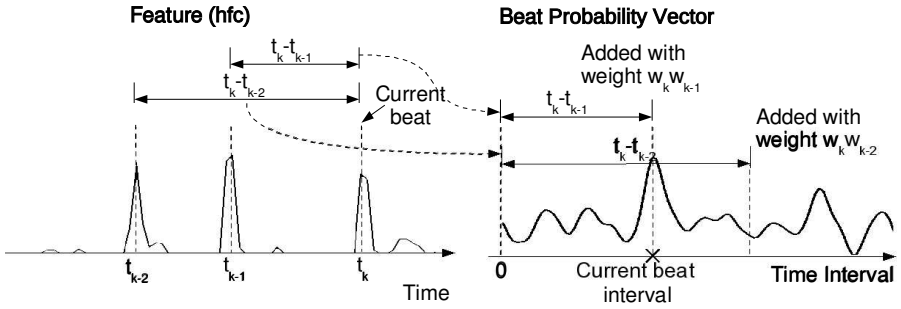

Fig. 3. *Selection of beats in the beat probability vector. For each new peak (left), a number of previous intervals are scaled and added to the vector (right). The maximum of the beat probability vector gives the current beat interval.*

4.2 Update with multiple intervals

To avoid a situation where spurious peaks create a maximum in the probability vector with an interval that does not match the current beat, the vector is updated in a novel way. By weighting each new note and taking multiple previous note onsets into account, the probability vector $H(t)$ is updated with N previous weighted intervals that lie within the allowed beat interval,

$$H(t) = H(t) + \Sigma_{i=1}^{N} w_k w_{k-i} G(t_k - t_{k-i}, t), t = 0 \ldots \infty \tag{14}$$

For simplicity, the time weight W is omitted in this formula.

This simple model gives a strong indication of note boundaries at common intervals of music, which permits the identification of the current beat interval.

An illustration of the calculation of the beat probability vector can be seen in figure 3. It consists of the estimated audio feature (left), the estimation of probable beat and the updating of the running beat probability vector (right). The current beat interval is now found as the interval closest to the maximum in the beat probability vector. If no such interval exists, the maximum of the beat probability vector is used.

5 Evaluation

The beat estimation has been evaluated by comparing the beat per minute (BPM) output of the algorithm to a human estimate. The human estimate was found by tapping along while the musical piece was playing, and finding the mean time difference between taps.

To evaluate stability of the algorithm 10 pieces of popular and electronic music was randomly selected from a large music database. In all cases the algorithm gave a stable output throughout the piece, after a startup period of 1 to 60 seconds. The long startup period is due to the nature of the start of these pieces, i.e. non rhythmic music. In six of the cases the estimated BPM value matched the human estimate, while in the remaining four cases, the algorithm estimate was half that of the human estimate. The problem of not estimating the right multiple of BPM is reported elsewhere [3], however, it is worth noting that in the case of controlling the tempo of the music, it is of primary importance to have a stable output.

In addition, informal use of the system in real-time audio conducting [7], DJ beat matching and tempo control [6] has shown that the beat estimation is stable for a large variety of music styles.

6 Conclusions

This paper presents a complete system for the estimation of beat in music. The system consists of the calculation of an audio feature that has been selected from a large number of potential features. A number of error measures have

been calculated, and the best feature has been found, together with the optimum threshold and block size, from the analysis of the error measures. The selected feature (high frequency content), is further enhanced in a beat probability vector. This vector, which keeps in memory the previous most likely intervals, renders an estimate of the current interval by the maximum of the beat interval probabilities.

The paper has presented several new features, a novel approach to the feature selection, and a versatile beat estimation that is both precise and immediate. It has been implemented in the DJ software Mixxx [14] and used in two well proven real-time music control systems: Conducting audio files [7] and DJ tempo control [6].

References

1. Goto, M., Muraoka, Y.: A real-time beat tracking system for audio signals. In: Proceedings of the International Computer Music Conference. (1995) 171–174
2. Goto, M., Muraoka, Y.: A real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. Speech Communication **27** (1998) 311–335
3. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. J. Acoust. Soc. Am. **103** (1998) 588–601
4. Dixon, S.: Automatic extraction of tempo and beat from expressive performances. Journal of New Music Research **30** (2001) 39–58
5. Desain, P.: A (de)composable theory of rhythm. Music Perception **9** (1992) 439–454
6. Andersen, T.H.: Mixxx: Towards novel DJ interfaces. Conference on New Interfaces for Musical Expression (NIME'03), Montreal (2003)
7. Murphy, D., Andersen, T.H., Jensen, K.: Conducting audio files via computer vision. In: Proceedings of the Gesture Workshop, Genova. (2003)
8. McAdams, S., Winsberg, S., Donnadieu, S., Soete, G.D., Krimphoff, J.: Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. Psychological Research **58** (1995) 177–192
9. Jensen, K., Murphy, D.: Segmenting melodies into notes. In: Proceedings of the DSAGM, Copenhagen, Denmark. (2001)
10. Dixon, S., Goebl, W., Widmer, G.: Real time tracking and visualisation of musical expression. In: II International Conference on Music and Artificial Intelligence. Volume 12., Edinburgh, Scotland (2002) 58–68
11. Beauchamp, J.: Synthesis by spectral amplitude and "brightness" matching of analyzed musical instrument tones. Journal of the Acoustical Society of America **30** (1982)
12. Masri, P., Bateman, A.: Improved modelling of attack transient in music analysis-resynthesis. In: Proceedings of the International Computer Music Conference, Hong-Kong (1996) 100–104
13. Gordon, J.W.: The perceptual attack time of musical tones. J. Acoust. Soc. Am. **82** (1987)
14. Andersen, T.H., Andersen, K.H.: Mixxx. http://mixxx.sourceforge.net/ (2003)