Technische Universität München
Lehrstuhl für Mensch-Maschine-Kommunikation
Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll

Diploma Thesis

Onset, Beat, and Tempo Detection with Artificial Neural Networks

Sebastian Böck

sb@minimoog.org

Abstract

A lot of different methods for onset, beat, and tempo detection have been proposed over the last years. But all of these have deficiencies. For onsets, there doesn't exist a single algorithm, which is capable of identifying onsets in all kinds of different sounds. Common drawbacks of beat and tempo detection algorithms are their dependency on high level knowledge or the assumption of a constant tempo throughout the entire analysed section.

In this work, a new approach with superior performance is introduced. It is based on bidirectional Long Short-Term Memory recurrent neural networks. It is completely data driven and doesn't require any higher level knowledge. With only a few modifications, it is general applicable for both onset and beat detection, the tempo is simply calculated on basis of the detected beats.

Speaking of the onsets detector, the neural network replaces the reduction function used in most traditional methods. Training of the network is performed on a large database of onset data covering various genres and onset types. Due to the data driven nature, the new approach renders the need to choose a special onset detection method and adjust its parameters depending on the music type obsolete. Beats detection is solely based on signal features, too. The new approach does not expect the beats to occur at strict metrical levels, and is hence capable of tracking beats even in the case of abrupt tempo changes.

Results are given for well known data sets, and it can be concluded that the new approach is on par with state-of-the-art methods and outperforms them in almost all cases. For onset detection of complex music with mixed onset types, an absolute improvement of 3.6% in terms of F-measure is reported. For tempo detection the performance gain for correctly identified tempo is 5.4% to 8.9% absolute, depending on the data set.

Contents

Introduction

Music is part of almost everybody's life. Contrary to spoken languages, music can be understood by people not speaking the same language. It connects people independently of their cultural background all over the world. Music can break down language barriers and thus helps understanding each other. This is because it also acts on a highly emotional level. If asked to describe music, people often use their own words and express their feelings.

But for a lot of applications, a more scientific characterisation is needed. Music is usually described by technical attributes, such as pitch, temporal, harmonic, timbral, editorial, textual, and bibliographic aspects [24]. This thesis deals with the temporal aspect of music, namely the two most fundamental features: onsets and beats.

Depending on the point of view, specific temporal events in music pieces are more relevant than others. For the typical listener, the most perceptible events in music are the beats. We nod our head or tap our foot to the beat of music. On the other side, for musicians or performers, the single note onsets are more important, since they correspond to the timings of the notes within the musical score. The automatic identification of all these events in polyphonic music is essential for a wide range of applications.

Precisely detected onsets are a prerequisite for audio to score transcriptions [11, 52, 74, 96]. They are further of great help for the automatic segmentation and analysis of acoustic signals, and can assist in typical cut-and-paste operations and help editing audio recordings [59].

Locating the beats and determining the correct tempo opens new possibilities for all kinds of music application tasks, such as automatic manipulation of rhythm, time-stretching, or swing modification of audio loops [42, 7]. Beats are also crucial for analysing the rhythmic structure, dancestyle [31, 83], and genre of songs [22, 40, 99, 68, 67], as well as identifying cover songs [29] or the similarity of songs [77, 56].

Deficiencies of existing methods

All existing methods for onset, beat and tempo detection (see section 1) have deficiencies. For onsets, there doesn't exist a single algorithm, which is capable of identifying onsets in

all kinds of different sounds [5, 13, 21]. These include pitched non-percussive music such as voices or bowed strings, pitched percussive sounds such as piano music, non-pitched percussive drum sounds, and complex mixes, including all types of onsets plus modern post processing effects (e.g. flangers and vocoders), which distort the signal considerably.

Common drawbacks of beat and tempo detection algorithms are their dependency on high level knowledge or the assumption of a constant tempo throughout the entire analysed section. E.g., many algorithms expect the beats occurring at strict multiples of the base metrical level (called the tatum) [84], work only on a given time signature [38], or require the existence of certain sound patterns [37]. Moreover, the best detection performance is in the range of 50-70%, depending on the data set [73], leaving room for large improvements.

Aim of this thesis

The aim of this thesis is hence to develop a universally applicable, purely data driven model for onset and beat detection in polyphonic music. It should achieve a very high level of correctness and precision, independently of the type of music. It should further not depend on any high level knowledge (e.g. the strict alignment of the beats on other metrical levels) or constant tempo.

Overview

This thesis is structured as follows: section 1 summarises existing methods for onset, beat, and tempo detection. After giving a short overview of artificial neural networks with a special emphasis on bidirectional Long Short-Term Memory neural networks (BLSTM) in section 2, a new approach for onset, beat, and tempo detection, based on BLSTM networks is introduced in section 3. The evaluation procedure is described in section 5, and results are given in section 6.

Chapter 1

Existing methods

This chapter gives an overview over existing methods for onset detection, tempo induction, and beat tracking. Methods with a close relation to the new approach, as well as state-of-the-art methods, are described into more detail.

1.1 Onset detection

An onsets corresponds to an event in the music signal. It indicates salient parts, such as changes in notes, chords, and all kind of different events like drum hits. It is important to distinguish between some related notions: onsets, transients, attacks, and decay. Figure 1.1 shows how these relate to each other, the definitions are given in the following.

Onset: The onset is defined as the start of a musical note or any other sound. For simplicity, the starting point of a transient (or the time at which a transient can be detected reliably by a human) is chosen as the onset of a sound. But soft harmonic sounds do not necessarily include a transient, so the onset position is harder to detect.

Transient: The transient is the short interval of the signal, which corresponds to the non-harmonic attack phase. Sometimes also the decay phase is considered to be part of the transient. During this phase, mostly unpredictable, non-periodic, high-frequency components can be observed, which are not related to the harmonic content of the underlying sound.

Attack: The attack is the part of the sound, where the amplitude of the envelope rises from zero to its maximum peak.

Decay: After reaching this peak, the amplitude of the envelope decreases to zero again. This phase is called the decay of the sound, and generally lasts much longer than the attack phase.

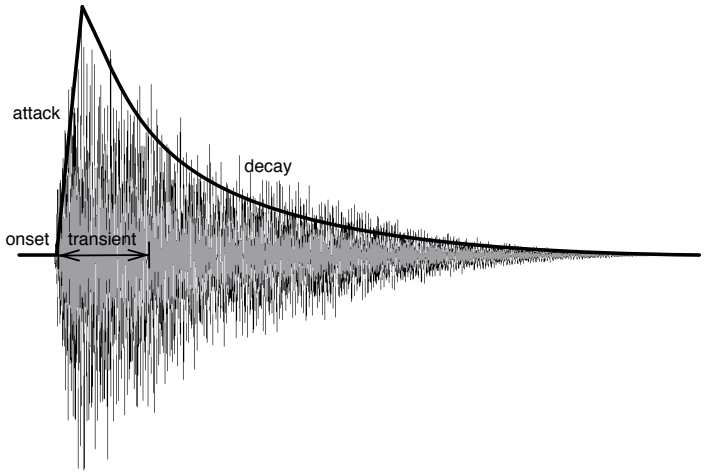


Figure 1.1: The relation of onset, attack, transient, and decay for a simple bell sound.

Historically, the first onset detection methods worked only with symbolic data, such as MIDI (Musical Instrument Digital Interface). In the mid 1990s first approaches arose, which took not only symbolic data, but also monophonic audio samples. This was a huge step towards onset detection in polyphonic music. The methods emerged more and more and are now sophisticated enough to handle even onsets in complex music mixes quite well [5, 12, 14, 21, 59, 62, 71, 92, 93, 97]. Nonetheless, none of the existing methods is capable of handling onsets in all kind of different sounds with the same performance.

The rest of the section describes the most common of these approaches capable of handling complex music mixes. Because of their close relation to the later described new approach, the signal feature based methods are described into more detail.

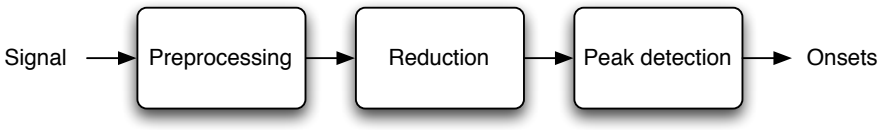


Figure 1.2: Basic workflow of traditional onset detection methods

Figure 1.2 shows the basic workflow of almost all traditional onset detection methods. The process is divided into the following steps:

Preprocessing: Some onset detection methods include a preprocessing step. The aim of preprocessing is to accentuate relevant parts of the signal. [93] describes a method called adaptive whitening, which attenuates irrelevant parts of the signal. Since preprocessing is not used for the new approach, it is not described further.

Reduction: A reduction function is applied to the (preprocessed) signal, to obtain the detection function. The detection function usually has a much smaller sampling frequency (around 100 Hz compared to the 44.1 kHz of the original signal). This reduces the complexity and informational content by a huge amount The reduction functions can be divided into two main categories, whether they achieve the reduction based on signal features or on probability models. Approaches of both categories are described later in this section, as well as some alternative approaches.

Peak detection: The task of this last stage is to reliably detect the onsets within the detection function. It is subdivided into post processing (e.g. smoothing and normalising of the detection function), thresholding, and peak picking. Thresholding is usually done with either a fixed or an adaptive threshold (e.g. a moving median, as used in [5]). Since the former one tends to pick either to much onsets in louder part, or miss onsets in quieter parts, usually adaptive thresholds are used. Finally peak picking is used to identify the local maxima.

The next sections describe some commonly used signal reduction functions.

1.1.1 Detection based on signal features

There are several methods for onset detection that use signal feature based reduction functions. Early methods operated directly on the music signal $x(n)$ in the time domain. One of them is the method described in [59]. The system first normalises the loudness of the signal before splitting it into multiple bands via bandpass filters. Peaks in the first order difference of the logarithm of the amplitude envelope of each band are detected as onsets. These band-wise onsets are then combined to yield the final set of detected onsets.

When operating in the time domain, quiet onsets are often masked by higher energy signals. Thus, a lot of reduction functions use the frequency domain representation of the audio signal for onset detection, in particular the short-time Fourier transform (STFT) $X(n,k)$ of the signal $x(n)$:

$$X(n,k) = \sum_{l=-\frac{N}{2}}^{l=\frac{N}{2}-1} w(l) \cdot x(l+nh) \cdot e^{-2\pi jlk/N} \tag{1.1}$$

where n is the frame index, k the frequency bin number, and $w(l)$ is the windowing function of size N.

High Frequency Content

In [72], a method called high frequency content (HFC) is described, which linearly weights each STFT bin with a factor proportional to its frequency. Summing all weighted bins yields a measure for the onset strength, which is used as a detection function.

$$HFC(n) = \frac{2}{N} \sum_{k=1}^{k=\frac{N}{2}} |k| \cdot |X(n,k)|^2 \tag{1.2}$$

When dealing with percussive music, this method reveals its strength, but shows weaknesses otherwise. Percussive onsets lead to wide-band noise in the spectrogram up to the highest frequencies, and this is exploited by the HFC method. But it does not incorporate the temporal evolution of the signal, as it considers only absolute energy values at the present frame.

Spectral difference

The spectral difference (SD) also uses the temporal information as it calculates the difference of two consecutive short-time spectra. This difference is built separately for each frequency bin, and all positive differences are then summed up to build the detection function. Some publications use the L_2-norm, also known as the Euclidean distance (equation 1.3) [27], whereas others use the L_1-norm (equation 1.4) [72].

$$SD(n) = \sum_{k=1}^{k=\frac{N}{2}} \{H(X(n,k) - X(n-1,k))\}^2 \tag{1.3}$$

$$SF(n) = \sum_{k=1}^{k=\frac{N}{2}} H(X(n,k) - X(n-1,k)) \tag{1.4}$$

with $H(x) = \frac{x+|x|}{2}$ as the half-wave rectifier function. The latter version is often referred to as the Spectral Flux (SF) [2, 64]. These methods are good overall performers, for almost any kind of music material.

Phase deviation

The methods mentioned so far used only the magnitude of the spectrum. More recent approaches [8, 6] utilise the phase of the signal's STFT. The change of the phase in a STFT frequency bin is a rough estimate of this bins instantaneous frequency. If $\varphi(n,k)$ is the phase of $X(n,k)$ with a range of $-\pi < \varphi(n,k) \leq \pi$, then the instantaneous frequency is given by the first order difference $\varphi' = \varphi(n,k) - \varphi(n-1,k)$. The change in the instantaneous frequency, the second order difference given by $\varphi'' = \varphi'(n,k) - \varphi'(n-1,k)$, is an indicator of a possible onset. To reduce the chance of an missed onset because of wrap around of the phase to 0, the phase deviation (PD) onset detection function takes the mean over all frequency bin phase changes:

$$PD(n) = \frac{2}{N} \sum_{k=1}^{k=\frac{N}{2}} |\varphi''(n,k)| \tag{1.5}$$

In [21], two improvements to the phase deviation onset detection function called weighted phase deviation (WPD) and normalised weighted phase deviation (NWPD) are proposed. The WPD function weights each frequency bin of the phase deviation function with its magnitude. This takes into account the fact that the energy is concentrated around the bins with the dominant frequency of the sounding tones, and thus makes this new detection function more robust against noise.

$$WPD(n) = \frac{2}{N} \sum_{k=1}^{k=\frac{N}{2}} |X(n,k)\varphi''(n,k)| \tag{1.6}$$

The NWPD function additionally normalises the WPD function with the sum of the weights and is defined as:

$$NWPD(n) = \frac{\sum_{k=1}^{k=\frac{N}{2}} |X(n,k)\varphi''(n,k)|}{\sum_{k=1}^{k=\frac{N}{2}} |X(n,k)|} \tag{1.7}$$

Complex Domain

Another way to incorporate both magnitude and the phase information (as in the last two detection functions) is proposed by [25] and [6]. First, the expected amplitude and

phase for the actual frame is estimated based on the two previous ones. The so called target value is calculated by the assumption of constant amplitude and rate of phase change as: $X_T(n,k) = |X(n,k)|e^{\varphi(n-1,k)+\varphi'(n-1,k)}$. Then this value is compared to the actual frame.

The complex domain (CD) onset detection function is defined as the sum of all deviations between the actual values and the calculated target values.

$$CD(n) = \sum_{k=1}^{k=\frac{N}{2}} |X(n,k) - X_T(n,k)| \tag{1.8}$$

A modification of this method called the rectified complex domain (RCD) deals with the problem that the original algorithm does not distinguish between increases or decreases of the signals amplitude [21]. Similar to the idea used in the spectral flux function, only the positive values of the complex domain are summed up.

$$RCD(n) = \sum_{k=1}^{k=\frac{N}{2}} H(X(n,k) - X_T(n,k)) \tag{1.9}$$

with $H(x) = \frac{x+|x|}{2}$ as the half-wave rectifier function.

Wavelet regularity modulus

Not taking the STFT spectrogram as the origin for the used features, but rather a time-frequency representation is the base for the wavelet regularity modulus (WRM) method [16]. It tries to detect transients in the Haar wavelet decomposition of the signal by observing regularities in the wavelet coefficients. High amplitudes in a certain coefficient often coincide with high amplitudes in coefficients at the same time localisation at smaller scales and therefore form structures. Detecting a transient also implies the detection of an onset (see figure 1.1), but the onsets of soft sounds without transients are hard to detect.

1.1.2 Detection based on probability models

Besides signal features, statistical methods can be used for onset detection as well. They are based on the hypothesis that a signal can be described by probability models.

The negative log.-likelihood (NLL) method [4] therefore defines two different statistical models and observes, whether the signal follows the first or the second model. A sudden change from the first model to the second one can be used as an onset detection function. However since the logarithm of the ratio between the two probabilities is used, onsets are not indicated as peaks, but as a change in sign. This method so far performs best for pitched sounds without any percussive components, e.g. bowed strings and vocals.

1.1.3 Machine learning algorithms

To build more sophisticated detection functions, which are capable of detecting onsets in a wider range of audio signals, classifier based data driven methods emerged lately.

In [71], a network of integrate-and-fire neurons and a multilayer perceptron (MLP) is used for onset detection in polyphonic piano performances. A similar onset detector is proposed in [92]: it simulates the human hearing by using the same neurons, but with depressing synapses. This algorithm was only tested against tone bursts, simple sounds and speech samples, but results for complex polyphonic music samples were missing.

Lacoste et. al. describe an onset detection algorithm based on a feed forward neural net, namely convolutional neural nets [61, 62]. This system performed best in the MIREX 2005 audio onset detection contest. Unfortunately no results on available test sets are published, only the ones of the contest, which are based on a publicly unavailable test set.

Besides neural networks, other machine learning algorithms such as Support Vector Machines (SVM) and Naive Bayes classifiers (NB) were proven to work well for onset detection, although obtaining only mediocre precision rates [12]. In [18], a time-frequency representation of the signal is combined with support vector machines to detect abrupt spectral changes, but no test results are included.

Originally developed for music transcription of piano music, the systems based on the non-negative matrix factorisation (NMF) of magnitude spectra [91] can also be used to extract onsets. One shortcoming of this approach, the requirement of sounds from instruments that exhibit a static harmonic profile, was resolved by the use of non-negative matrix deconvolution (NMD) [90]. However, no comparable results are given.

1.2 Tempo induction and beat tracking

For humans, tracking the beat is an almost natural task. We tap our foot or nod our head to the beat of the music. Even if the beat changes, humans can follow it almost

instantaneous. Nonetheless, for machines the task of beat tracking is much harder, especially when dealing with varying tempi, as the numerous publications by different authors on this subject suggests [1, 2, 17, 19, 29, 31, 37, 43, 51, 64, 80, 81, 82, 84, 86, 89, 101, 102]. Depending on the structure of the implemented method, they have to make certain assumptions about the music signal. Some methods need the beats to occur at strict metrical levels, while others rely on certain sound patterns like drum sounds. The new approach described in chapter 3 lifts all these limitations.

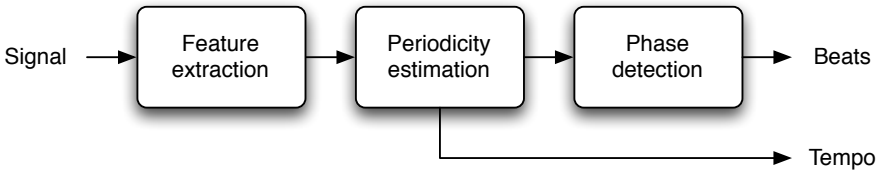


Figure 1.3: Basic workflow of traditional tempo induction and beat tracking methods.

The rest of this section describes the most widely used algorithms for beat tracking and tempo induction. Most methods have a working scheme like the one shown in figure 1.3. After extracting features from the audio signal, they try to determine the periodicity of the signal (the tempo) and the phase of the periodic signal (the beat locations).

The complete process is divided into the following steps:

Feature extraction: The input signal is processed and certain features are extracted. These features can be onsets (see previous section), rhythmic informations, chord changes, amplitude envelopes, spectral features, etc.. The choice mostly depends on the pulse induction stage used in the next step.

Periodicity estimation: The purpose of this stage is to determine the periodicity of the most dominant pulses in the extracted features. This stage is the core of all tempo induction algorithms. As a result of this step, the tempo can be calculated.

Phase detection: Some methods also produce phase informations during periodicity estimation, and therefore don't need this last step for beat tracking. All others need to determine the phase of the periodic signal, i.e. the position of the pulses.

For periodicity estimation, the autocorrelation, comb filter, histogram, and multiple agent based induction methods are widely used. These are described more detailed in the following sections.

1.2.1 Autocorrelation based methods

One of the most common methods for periodicity estimation is the autocorrelation function (ACF). It is a measure for the self-similarity of a function with itself at a certain delay τ. As the underlaying function, different methods can be used. To determine the most dominant tempo, the autocorrelation of this function is build, which exhibits the highest amplitude at a special delay τ^*. This delay is then converted to the tempo, typically given in beats per minute (bpm).

As mentioned before, different functions can be used for the autocorrelation. In [2], a special onset detection function, which is basically the spectral flux onset detection method of a time-frequency representation of the signal, is used for the ACF. To only capture the dominant onsets, it uses a much higher threshold than typically used for onset detection.

Another method that uses autocorrelation for tempo detection is described in [23]. This algorithm works on the signal envelopes of eight frequency bands. For each band, the signal is first rectified, down sampled, smoothed, and finally transformed to a logarithmic scale. Then the ACF is build for each frequency band and the results are combined.

The ACF only determines the periodicity of the signal. If not only the tempo but also the beat positions are needed, the phase of the pulses have to be discovered in a second step. For this purpose, the later described comb filter method can be used. Another possibility is to use Phase-Locked-Loop (PLL) techniques [89] for beat tracking. This method was later modified with a Kalman filtering approach [88, 87].

Some methods, which base on modifications to the traditional autocorrelation of pulse functions, have been proposed as well. One is described in [28]. It computes the autocorrelation such that the distribution of energy in phase space is preserved. The result is called a autocorrelation phase matrix (APM). Individual APMs over short overlapping onset traces are calculated, and the beats (and meter) are finally estimated by a Viterby decoding algorithm, which searches for points with smoothly changing lag and phase over time.

1.2.2 Histogram based methods

Another major approach is building a histogram of event intervals. The difference to the autocorrelation approach lays in the modality the histogram is built. Contrary to the ACF, the histogram methods does not consider the amplitude values of the function (or by other means: uses a function containing only values of 0 and 1). It simply sums up

the quantity of certain interval lengths. Typically the inter onset interval (IOI) of the detected onsets is used.

Histogram based methods usually perform worse than the similar working ACF based methods [23, 44], and are thus not described more detailed.

1.2.3 Comb filter based method

A third widely used method for periodicity estimation is the comb filter based method introduced by Scheirer in [81]. The basic idea is that a signal is passed through a set of comb resonators, and to measure the sum of magnitudes at the output. If the resonance frequency of the filters match the signal pulses, a high output can be observed. The main advantage compared to the autocorrelation based method is that it also able to track multiples and divisors of the given rate. Besides the complexity for implementation (the number of comb filter banks needed to cover the whole tempo range equals to the number of delay steps for the periodicity length), it's major drawback is that care hast to be taken, if the tempo varies. Since this method processes the signal sample by sample, it can be used directly for beat tracking purposes, by simply observing the comb filters output. Therefore the additional phase detection stage is not necessary.

The original implementation [81] operated in the time domain. It splits the signal into six bands, and each band is passed through a bank of tuned resonators. The output of the resonators are combined to get the final tempo.

Klapuri enhanced this algorithm substantially by processing the signal not in its temporal, but rather in a time-frequency representation [60]. Like the original version, it splits the signal into multiple bands, but uses a much larger number of bands (36), so that harmonic changes can be detected too. Since it is not meaningful to predict the periodicity on each of these sub bands, they are recombined to four accent bands by summing their power differences across frequency ranges and transforming them to a logarithmic power scale, before analysing the periodicity. By adding a Bayesian probabilistic model, this algorithm is further able to estimate not only the tactus (beat), but also the tatum (the smallest temporal unit [84]), and the measure of the song. This algorithm won the MIREX 2004 tempo induction contest.

Schuller et. al. describe an algorithm, which is basically the same as the original implementation, but with a few modifications [83]. It extends the search range for the comb filter bank to include higher metrical levels as well. The extracted low level features are used to detect higher level features, such as meter and dance style, by a support vector machine (SVM). Based on this additional dance style information, the relevant tempo range can be in turn constrained further, resulting in a much better tempo prediction.

It clearly outperforms the before mentioned winner of the MIREX 2004 tempo induction contest on the Ballroom data set.

In [17], a two state model, which gets the beat periodicity and timing information of an onset detection function by applying adaptively weighted comb filter banks, is described.

1.2.4 Multiple agent based methods

Goto introduced the multiple agent architecture approach to beat tracking of music with [37] or without drum sounds [38]. It combines heuristic and correlation techniques to build beat hypotheses, which are then considered by multiple agents. Those agents choose the winning hypothesis and the final beat phase and tempo is determined. By not only including onsets and beat patterns, but also chord changes this system was further refined in [39, 36].

The idea of multiple agents, following different beat hypotheses, can be used in conjunction with other techniques as well. In [80], a multi-agent algorithm for beat and tempo analysis is proposed, which tries to combine the onset information with note accentuations to improve over the purely onset based multi-agent algorithms. The method described in [20] uses an onset detection function to build up clusters of inter onset intervals to find likely tempi and phase hypotheses, for which agents are created and destroyed over time.

1.2.5 Other methods

Besides the before mentioned algorithms, a lot of other methods have been proposed by different authors. In [3], a system for tempo estimation based on a harmonic and noise decomposition of the audio signal is introduced. Also adaptive learning techniques are incorporated to beat and tempo analysis [35].

A periodicity estimation algorithm based on combined temporal and spectral representations of the musical signal is proposed in [78], and in [85], wavetable oscillators for beat tracking.

Onsets are often used as a helper function for the determination of beats. They can be combined with a maximum likelihood estimator for locating beat positions [63]. Another system that uses onsets as a helper signal and then uses a particle filtering algorithm to associate these events to beats and extract a tempo, is described in [51]. If the beat spectrum [34, 33], which characterises the rhythmic structure of a musical piece, is combined with an onset detection function it could not only be used for tempo estimation,

but also for beat tracking. In [54], a self-adjusting beat detection and prediction system based on a recurrent timing network which takes an onsets stream as input is described. It achieves very mixed results, ranging from only 46% to 100% depending on different kinds of music material.

Most algorithms use signal features of the uncompressed signal, but there also exist models which base their tempo inductions algorithms on features already present in symbolic music representations such as MIDI [96], or compressed MP3 music files [101, 15].

Chapter 2

Artificial Neural Networks

Artificial neural networks show good results in a wide range of areas, from pitch [66] and key detection [94], head and stem recognition of notes [75], to complete music transcription [70]. Some of the best result ever reported for onset detection are based on a neural network approach [61, 62].

The basic structure of an artificial neural network (ANN) is a network of nodes which are connected with weighted connections. Their counterparts in the biological world are neurons and the synapses of a certain strength between them. The input provided at the input nodes activates the network, and this activation then spreads throughout the whole network along the weighted connections. The activations of the output nodes can be used for different tasks, such as classification of the inputs.

A lot of different neural network types and technologies have been proposed ever since their first appearance in the early 1940s. This chapter gives a short overview and introduction to the subject.

2.1 Neural network types

This section describes the most basic neural network types and categorises them. Whatever network type is chosen, the basic structure and workflow is the same.

The input nodes provide their activations to all connected nodes. These nodes are usually modelled as simple summarisers, which calculate a weighted sum of all input values of the units connected to them. An activation function is then applied to this sum, yielding the final activation of the unit. As activation functions a simple linear or threshold function can be used. More commonly, however, are two nonlinear functions, the hyperbolic tangent (output values between -1 and 1) or the logistic sigmoid (output values between 0 and 1). Both map an infinite input value range to a defined finite output range.

Depending on the task of the network, the number of used output units and their activation functions have to be chosen accordingly. E.g., for binary classification a single output unit with the logistic sigmoid activation function is used. If more than two classes are needed, usually one output unit for each class is used, and a soft-max function is used to obtain the class probabilities.

2.1.1 Feed forward neural networks

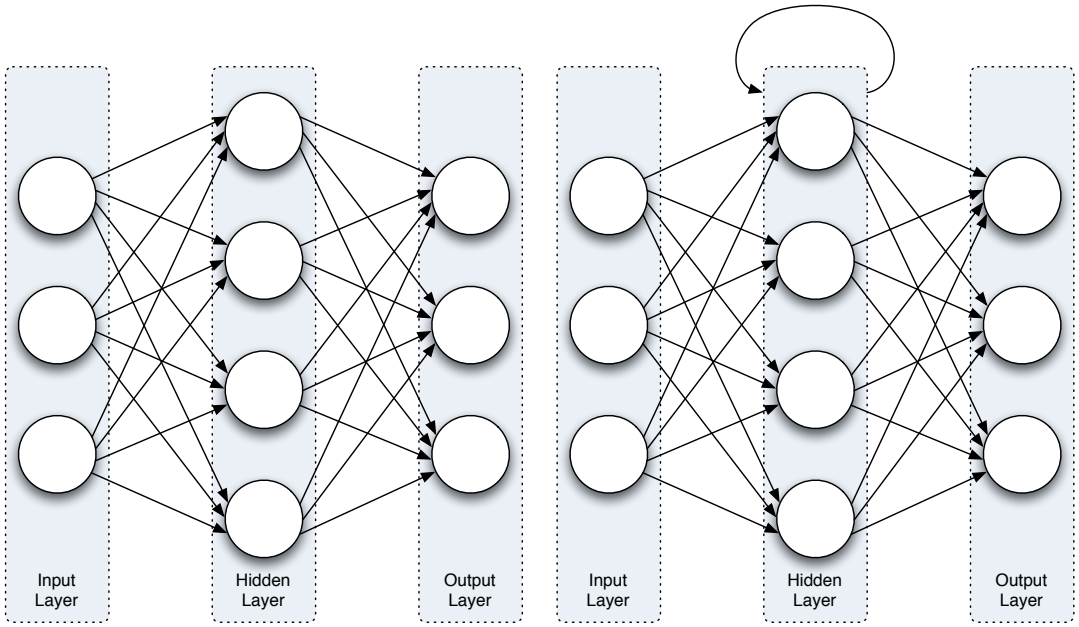
Neural networks, whose connections between nodes are directed forward only from the input nodes to the output nodes, are called feed forward neural networks (FNN). The most common form of a FNN is the multilayer perceptron (MLP) [79]. It consists of a minimum of three layers, one input layer, one or more hidden layers, and an output layer, as shown in figure 2.1(a). The connections feed forward from one layer to the next without any cycles. Contrary to the standard linear perceptron, the hidden units of MLPs have nonlinear activation functions, typically the hyperbolic tangent or the logistic sigmoid. FNNs are causal systems, because the output values solely depend on the input values. This type of network is suitable for pattern classification. Using as many output nodes as there are classes, MLPs with logistic sigmoid units are widely used as classifiers, by simply choosing the output with the highest activation.

2.1.2 Recurrent neural networks

Networks, whose connections between nodes form cycles, are called recurrent neural networks (RNN). The easiest form of a RNN is the one shown in figure 2.1(b), where only the hidden layer is connected to itself. A lot of different approaches of how these cycles in RNNs can be built, were proposed. Since the basic concepts are the same for all RNNs, they are not described further.

The biggest difference between FNNs and RNNs is that a RNN can theoretically map from the entire history of previous inputs to an output. The recurrent connections form a kind of memory, which allows input values to persist in the hidden layer of the network and therefore influence the network's output at a later time.

If not only the past, but also the future context of the input is necessary to determine the output (as in many real world examples like letter classification), there are different strategies to circumvent this shortcoming. One is to add a fixed time window to the input, and thus have access to information both in the past and the future. Another solution is to add a delay between the input values and the output targets. Both measures have their downsides. First, the size of the input layer is increased reasonably, and second, the input values and output targets are displaced.



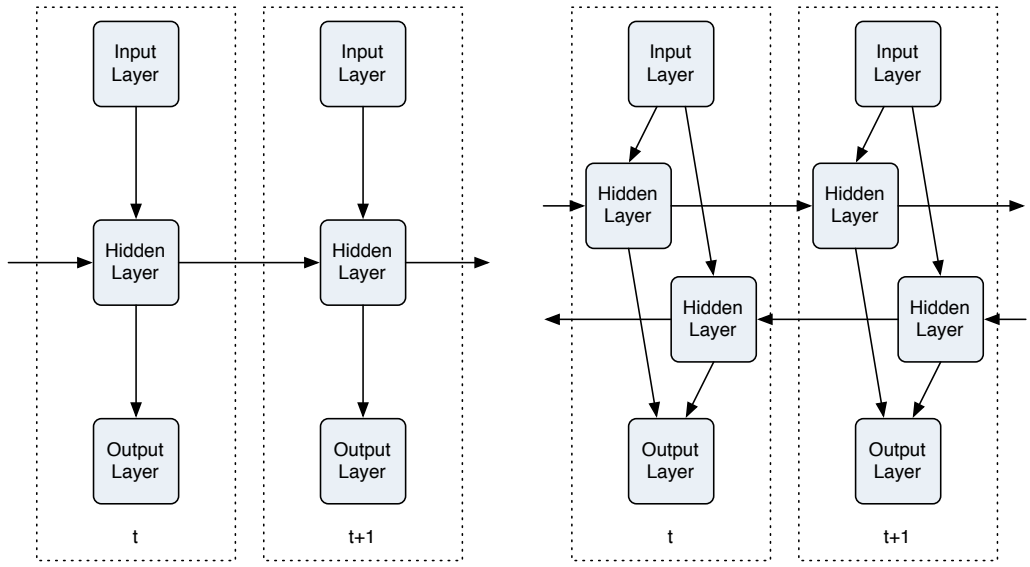
(a) Feed forward neural network (b) Recurrent neural network

Figure 2.1: Feed forward and recurrent neural networks

2.1.3 Bidirectional recurrent neural networks

Bidirectional recurrent networks (BRNN) offer a more elegant solution to the problem, as they not only provide access to past input values, but also to future inputs. Two separate hidden layers are used instead of one, both connected to the same input and output layers (see figure 2.2). Inputs values and the corresponding output targets are presented to those hidden layers in a forward and a backward state. Therefore, the complete network always has access to the complete past and the future context in a symmetrical way, without bloating the input layer size or displacing the input values and the corresponding output targets.

Besides the mentioned advantages, BRNNs have a major drawback, too. They violate causality. For certain tasks, such as financial prediction, this is not feasible. But for a lot of common problems, causality is not mandatory. If the inputs are spatial and not temporal this is obvious (and can bee seen in the dominance of BRNNs for bioinformatic tasks such as protein structure prediction). Also, if the input is of temporal type, but the output is not needed instantaneously, BRNNs can be applied successfully. A lot of



(a) RNN (b) Bidirectional RNN

Figure 2.2: Standard and bidirectional recurrent neural networks

speech or handwriting recognition tasks fall into this category. Even for real-time tasks BRNNs can be used, as long as the input can be split into chunks and it is acceptable to wait for the result until the end of each part.

2.2 Long Short-Term Memory

Although (bidirectional) recurrent neural networks have access to past (and future) information, the range of context is limited. The problem lays in the fact that the influence of an input value decays or blows up exponentially over time (depending on whether the weights are greater or lesser than 1), as it cycles through the network with its recurrent connections. This problem is called the vanishing gradient problem, and limits the range of contextual information accessible by RNNs to as few as ten time steps between the relevant input and the corresponding output events.

Hochreiter et. al introduced a novel and efficient gradient-based method called "Long Short-Term Memory" (LSTM) to overcome this deficiency [55] . In the LSTM terminology, a unit of recurrently connected subnets is called a memory block. Figure 2.3 shows such a LSTM memory block. Each block contains one or more self connected linear

memory cells (orange), three multiplicative units (violet), and three gates (green).

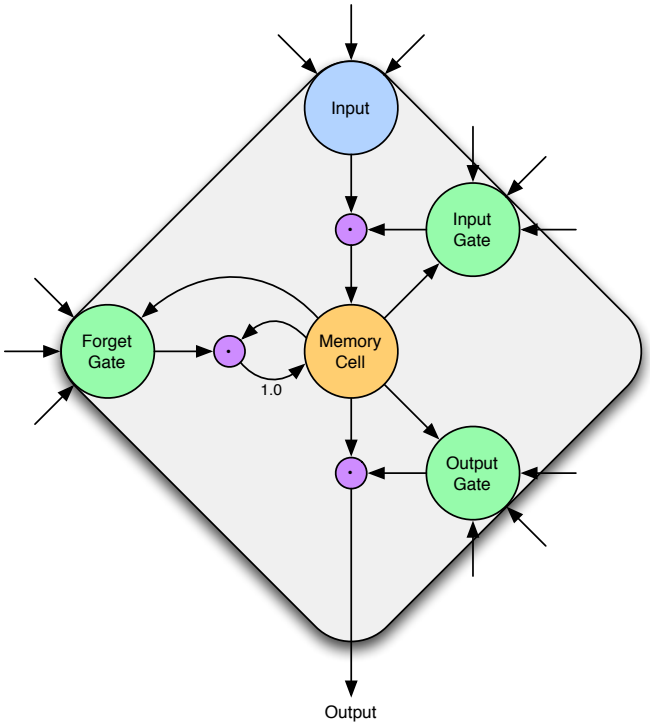


Figure 2.3: Long Short-Term Memory block with one memory cell

The internal state of the cell is maintained with a recurrent connection with a fixed weight of 1.0. In the analogy of computer memory cells, the three input, output, and forget gates correspond to write, read, and reset operations. The gates allow a LSTM memory cell to store and access informations over long time periods. New input values are stored in the cell only if the input gate opens (i.e. has a value close to 1). Stored values are accessible by the network if the output gate opens, and as long as the forget gate is open, the internal state is kept.

We speak of a bidirectional Long Short-Term Memory (BLSTM) network, if the non-linear units in the hidden layer of a bidirectional recurrent neural network are replaced by LSTM memory blocks. A more detailed description of the LSTM architecture, including the forward and backward passes needed for training, can be found in [45].

2.3 Network training

In general, three main techniques exist for machine learning. Supervised learning provides the classifier to be trained with pairs of input and target values. These pairs are named (i, t), with i being an element of the input space I and t being an element of the target space T. The other two main training methods are reinforced learning, where a positive or negative reward is provided to the classifier for training, and unsupervised learning, where no task specific training data is available, and the algorithm has to unveil the data's structure by inspection. In this thesis only supervised learning is used.

The aim of supervised learning is to find a function to map the input values as close as possible to the targets. The output of that function can be either of continuos values (regression) or one of several class labels (classification). During evaluation, the results for classification were always better than the ones achieved with regression. Since all targets (onsets and beats) can be mapped to classes, only the classification task is used.

To determine how successful the input values are approximated to the target values, the distance of these two values is used as the error, called the cross-entropy error. This error is then tried to be minimised with error backpropagation, using the gradient descent learning algorithm. A prerequisite for this learning algorithm is that all used activation functions in the units are differentiable. The learning procedure is divided into these steps:

Weight initialisation: all weights of the neural network are initialised with small random values. A Gaussian distribution with a mean of 0 and a standard deviation of 0.1 is used.

Forward pass: with the actual weights of the network and the current input values, the errors at the output node(s) of the network are calculated.

Backward pass: during this pass, all weights of the network are updated, so that the errors at the output nodes become smaller.

The forward and backward passes are repeated as long as the stopping criterion is not satisfied. Possible criterions are a maximum overall error or a fixed number of epochs used for training.

For supervised learning, the complete set S of input-target pairs (i, t), is split into two disjoint sets. The training set S_{train} is used for training, while the test set S_{test} is used solely to test the performance of the trained neural network on basis of data, previously unknown to the net. The ability of a network to transfer its performance from the training set to the test set is called generalisation. To achieve a high level of generalisation, different methods exist, with one of the simplest being early stopping.

Early stopping

For early stopping, which is used in this thesis, a third disjoint set, the validation set S_{val} is needed. This set is used to evaluate the performance of the neural network, and decides when to stop training in order to prevent over-fitting to the training data (see figure 2.4). This evaluation is usually performed at the end of each training epoch after calculating the errors for the validation set. At the beginning of the training, the errors for all data sets decrease. After an indefinite number of epochs however, the error for the validation and test sets begin to raise again, as the network adapts more and more to the training set. The network with the smallest cross-entropy error for the validation set is chosen as the winning network. This point does not necessarily coincide with the optimal point for the test set (which is located at epoch number 50 in figure 2.4), but should be close to it in most cases.

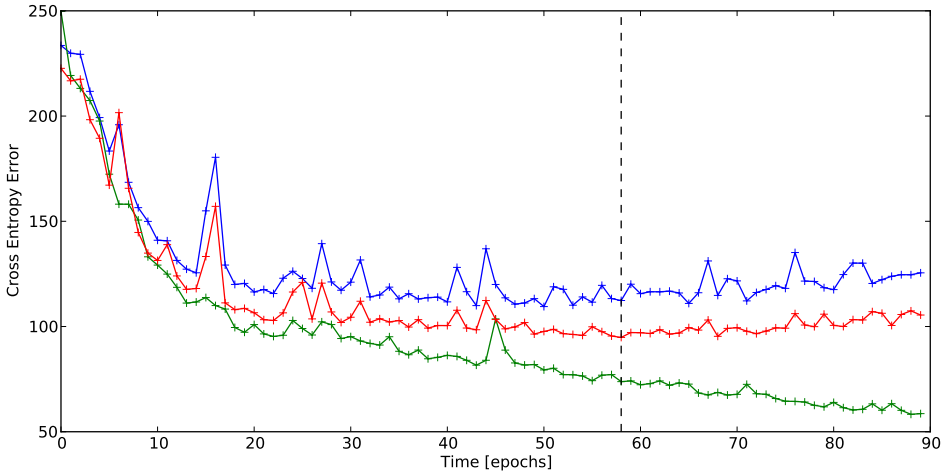


Figure 2.4: Learning curves with cross-entropy errors for the training set (green), validation set (red), and test set (blue). First, the errors decrease for all sets, before it increases again for the validation and test set. Marked with a vertical dashed line is the best validation network, determined by early stopping. The learning curves were recorded during the training of an onset detector.

Early stopping is a very reliable method to improve generalisation, but is has some drawbacks too. One being the fact that part of the set has to be used for validation, and is therefore not available for training. Another problem is that the chosen validation set might not be a good predictor for the later achievable performance of the test set.

Input representation

As with all machine learning tasks, choosing a suitable input representation is crucial for good performance. The input data therefore has to be both complete and reasonably compact [45]. Although neural networks are relatively forgiving the wrong choice of input representation (e.g. can sort out irrelevant information), high dimensional input vectors are undesirable. They lead to an unnecessary high number of input weights and hence the networks suffers bad performance and long training times.

A lot of evaluations have therefore been performed to determine a good input representation for the detection of onsets and beats. They are described in more detail in sections 5.1.1 and 5.2.1. The impact on the achievable performance is quite remarkable, it is around 5-10% absolute.

Chapter 3

New approach

The proposed approach is based on a BLSTM network. Compared to other neural network methods, the exceptional performance of BLSTM networks for phoneme classification [49], unsegmented sequence data labelling [46], and handwriting recognition [47, 48] seems to make these networks predestined for the use in music information retrieval tasks as well.

The new approach is completely data driven. It does not include any higher level knowledge (like rhythmic patterns) or constrains (like a certain time signature) of the signal. It is therefore generally applicable to all investigated tasks, namely onset, beat, and tempo detection.

This chapter describes the theoretical background of the implemented system, chapter 5 details all performed evaluations and tests to determine all parameters. Finally, in chapter 6, results are given for the new approach, comparing it to existing methods.

3.1 Onset detection

The implementation of the neural net based onset detector has a lot in common with the signal feature based onset detection methods. The basic workflow is shown in figure 3.1. As input, a raw PCM signal with a sampling rate $f_s = 44.1\,\text{kHz}$ is used. To reduce the computational complexity, the stereo signal is converted into a monaural signal by averaging both channels. This signal is fed through a signal processing stage before entering the BLSTM network. The output of the neural network is then further processed to obtain the final onsets. The following subsections describe each of these stages into more detail.

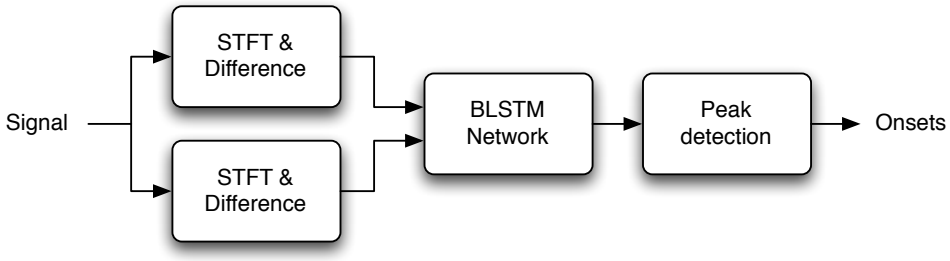


Figure 3.1: Basic workflow of the new onset detection method

3.1.1 Signal processing

Since onsets events are often masked in the time domain by higher energy signals [64], a common approach is to use the frequency domain instead [36, 27, 2, 5]. It has proven that this kind of processing outperforms direct temporal waveform processing.

The discrete input audio signal $x(n)$ of length L is therefore segmented into frames of N samples length. A finite windowing function $w(l)$ with a size of N samples is applied to each frame before analysing the signal. In [30], different windowing functions (i.e. half-wave sine and Hamming windows) were investigated, but it was concluded that the role of the window function is of less importance. Thus a standard Hamming window function $w(l)$ with a value of $\alpha = 0.46$ is used.

$$w(l) = (1 - \alpha) - \alpha \cdot \cos\left(\frac{2\pi l}{N}\right) \tag{3.1}$$

Depending on the chosen frame rate f_f (a value of 100 fps is used throughout this thesis), these windows are f_s/f_f samples apart (called the hop size h) and overlap each other with a factor of $(N - h)/N$. The short-time Fourier transform (STFT) $X(n, k)$ is computed as:

$$X(n,k) = \sum_{l=-\frac{N}{2}}^{l=\frac{N}{2}-1} w(l) \cdot x(l + nh) \cdot e^{-2\pi jlk/N} \tag{3.2}$$

where $x(n)$ denotes the signal, $w(l)$ the analysis window, h the hop size or time shift in samples between adjacent frames, n the frame index number, and k the frequency

bin number. In the ongoing, the phase information of the STFT is omitted and only the magnitude values are taken. This has the advantage that only real values instead of complex ones need to be handled. The spectrogram $S(n, k)$ is given by the squared magnitude of the STFT of the function:

$$S(n, k) = |X(n, k)|^2 \tag{3.3}$$

The own evaluation and other works [58] have shown that spectrograms with a better frequency resolution help to detect soft onsets. Since the spectrogram is used as the input for the neural network (and the spectrogram size increases linearly with the used window length), the drawback of incrementing the window size is a higher computational complexity. To reduce the size of the spectrogram, a conversion to the Mel frequency scale is performed. The Mel scale is a perceptual scale of the pitch of a tone and has a linear frequency response below 1127 Hz and a logarithmic scale above this frequency.

The Mel spectrogram is the matrix product of the spectrogram and the Mel filter bank $F(m, k)$ (for its computation, please refer to appendix A). During evaluation the use of $m = 40$ filters for onset detection has proven well to reduce the size of the spectrograms without a noticeable loss of performance. The Mel filter bank covers almost the complete human frequency range of 20 Hz to 16.0 kHz. To better represent the human perception of loudness, the Mel spectrogram $M(n, m)$ is given with logarithmic magnitudes:

$$M(n, m) = \log\left(S(n, k) \cdot F(m, k)^T + 1.0\right) \tag{3.4}$$

Additionally the first order differential $D(n, m)$ of the logarithmic Mel spectrograms is computed. It indicates a raise or reduction of the energy for each frequency bin at a frame relative to its predecessor.

$$D(n, m) = M(n, m) - M(n - 1, m) \tag{3.5}$$

Motivated by the procedure taken to compute the Spectral Flux (section 1.1.1), which is one of the best performing onset detection algorithms [5, 21], additionally the positive first order difference $D^+(n, m)$ is calculated by applying a half-wave rectifier function $H(x)$:

$$D^+(n, m) = H\left(D(n, m)\right) \tag{3.6}$$

$$H(x) = \frac{x + |x|}{2} \tag{3.7}$$

Traditional signal feature based methods for onset detection feed these signal informations through the reduction functions described in chapter 1. In contrast, the new method feeds the information into the neural network.

3.1.2 Neural Network

The neural network used is a bidirectional recurrent network with three hidden layers and 20 Long Short-Term Memory units each. Evaluation (section 5.1.1) has shown that using two spectrograms with window lengths of 23.22 and 46.44 ms (STFT window sizes of $N = 1024$ and $N = 2048$ with the used sampling rate $f_s = 44.1\,\text{kHz}$ and frame rate $f_f = 100\,\text{fps}$) and their positive first oder differences gives the best result for onset detection. In the ongoing the window size is used as an index, naming the spectrograms M_{23}, M_{46}, D^+_{23}, and D^+_{46} respectively. This data is used to construct the input vector i for the neural network, containing all the values of the above mentioned spectrograms for each frame n.

Usually, an improved performance can be observed by normalising the input values to have a mean of 0 and a standard deviation of 1 [65]. This normalisation doesn't change the input information by itself, but puts the values into a range more suitable for the activation functions used in neural networks. However, no performance gain could be achieved by this measure with the used input representations, therefore no normalisation step is performed.

Depending on the task, whether the network should be trained or tested, the target data (the onsets) has to be presented to the network as well. For training, the targets are therefore associated with one of the two possible classes, o and $\overline{o}$ (onsets and not onsets). The resulting target vector t has the same number of instances as the input vector i. The process of training is described into more detail in section 2.3. If an already trained network should be tested with previously unknown data, it is necessary to further process the output of the neural network, as described in the following section.

3.1.3 Peak detection

For onset detection the standard method for determining the class affiliation by choosing the output node with the highest activation value is not feasible, since a lot of onsets have activation values below this threshold (0.5 in case of the two used classes). Hence

the activation function $a_o(n)$ of the target class o is used for onset detection, but it needs to be processed further. The peak detection is straight forward and divided into the simple steps of thresholding, peak combining, and peak picking.

Thresholding

In traditional signal feature based onset detection methods, the detection function is always correlated with the signal, and therefore exhibits drastic changes in magnitude. This effect cannot be spotted with the new approach. In a sense, the neural network stage acts as a decoupler. The main advantage is that the activation values do not depend on the loudness of the signal. The actual activation value can be seen more as a level of certainty, whether an onset is present or not. Therefore a fixed (per song) threshold θ_o is used instead of an adopting one, and is calculated as:

$$\theta_o^* = \lambda_o \cdot \text{median}\{a_o(0), \ldots, a_o(L)\} \tag{3.8}$$

$$\theta_o = \min\left(\max\left(0.1, \theta_o^*\right), 0.3\right) \tag{3.9}$$

with $a_o(n)$ being the output activation function of the neural network for the onset class o, λ_o a factor determined experimentally on the validation set used for early stopping during the training of the neural network, and L the length of the audio signal. The λ_o^* yielding the best F-measure (for the used measures for evaluation and testing, please refer to section 4.5) is chosen. This threshold is then adjusted, to be within certain upper and lower bounds, which evolved during the evaluation process. The resulting onset function $o_o(n)$ containing only activation values $a_o(n)$ within this range and above the calculated threshold θ_o is given by:

$$o_o(n) = \begin{cases} a_o(n) & \text{for } a_o(n) > \theta_o \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

Peak combining

The onset function has a very high precision in case of suitable input data. Certain tasks, such as music transcription, require highly accurate onsets, whereas in other situations such a high precision is not needed or wanted, e.g. when onsets should be marked as they would be perceived by a human listener.

Research has shown that two onsets are heard synchronous up to a difference of 30 ms in the case of two tones and up to 70 ms if more tones are involved [53]. Asynchronies in the range of 30-50 ms are common in ensemble performances [95]. Hence peaks can be combined before the final peak picking step.

Depending on the used frame rate f_f and the chosen combination width c_w (in ms), the onset function $o_o(n)$ is convolved with a rectangular window r of appropriate size:

$$o^*_{o,comb}(n) = o_o(n) \star r \left(\frac{f_f \cdot c_w}{1000} \right) \tag{3.11}$$

Before proceeding with peak picking, only the centroids of the resulting combined peak clusters greater than 0 are used:

$$o_{o,comb}(n) = \begin{cases} \frac{\sum_{c=n_s}^{c=n_e} o^*_{o,comb}(c)}{n_e - n_s} & \text{for } o^*_{o,comb}(n_c) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

with n_c being the centre frames of the peak clusters, determined by the start (n_s) and end (n_e) frame calculated as:

$$n_s = n \quad \text{if } o^*_{o,comb}(n-1) = 0 \text{ and } o^*_{o,comb}(n) \geq 0 \tag{3.13}$$

$$n_e = n \quad \text{if } o^*_{o,comb}(n) \geq 0 \text{ and } o^*_{o,comb}(n+1) = 0 \tag{3.14}$$

$$n_c = \frac{\sum_{c=n_s}^{=n_e} c \cdot o^*_{o,comb}(c)}{\sum_{c=n_s}^{c=n_e} o^*_{o,comb}(c)} \tag{3.15}$$

Peak picking

Locating the local maxima in the (combined) onset function is the task of the peak picking step. An peak is considered an onset, if the following condition is met:

$$o(n) = \begin{cases} 1 & \text{for } o_o(n-1) \leq o_o(n) \geq o_o(n+1) \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

If the peaks were combined in the previous step, the onset function $o_o(n)$ in equation 3.16 needs to be replaced by the combined onset function $o_{o,comb}(n)$.

3.2 Beat detection

The beat detection workflow is almost identical to the one used for onset detection, and is shown in figure 3.2. It consists of the same stages, which are slightly modified. These modifications are outlined in the following subsections.

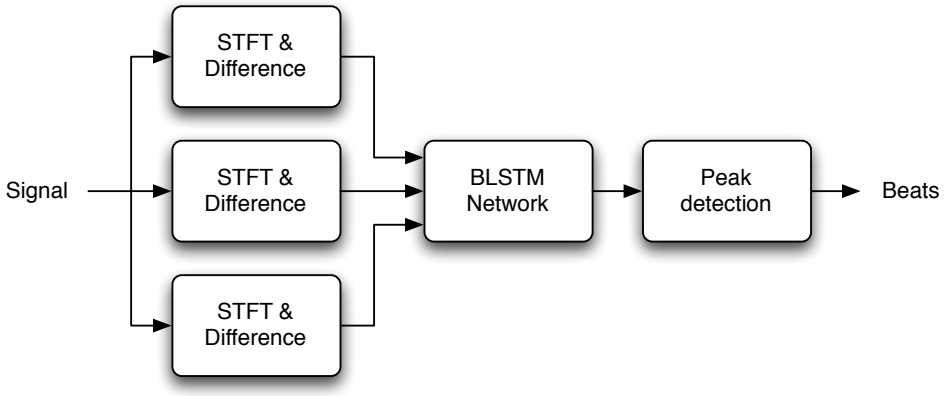


Figure 3.2: Basic workflow of the new beat detection method

3.2.1 Signal processing

The main difference in the signal processing stage is that the Mel filter bank only has 20 banks instead of 40. Another modification was made at the positive first order difference calculation step. A more sophisticated mechanism is used for beat detection. First the median average over a certain window of the spectrogram is taken, and then the first order difference is calculated against this moving average. The median average spectrogram $M^m(n,m)$ can be obtained according to:

$$M^m(n,m) = \text{median}\{M(n-l,m),\ldots,M(n,m)\} \tag{3.17}$$

with l being the length for the median average window. This length depends on the used window size N for the short time Fourier transform, and calculated as: $l = N/100$. The first order median difference $D^m(n,m)$ and its positive version $D^{+m}(n,m)$ are calculated according to these equations:

$$D^m(n,m) = M(n,m) - M^m(n,m) \tag{3.18}$$
$$D^{+m}(n,m) = H\left(D^m(n,m)\right) \tag{3.19}$$

with $H(x)$ being the known half-wave rectifier function already given in equation 3.7.

3.2.2 Neural Network

The network layout is very similar to the one used for onset detection, consisting again of a bidirectional recurrent network with three hidden layers, but in this case with 25 LSTM units each. Beat detection benefits from the use of longer analysis windows for the STFT. Therefore as inputs, a third spectrogram with a window length of 92.88 ms ($N = 4096$ frames) is used in addition to the two used for onset detection with window lengths of 23.22 and 46.44 ms. As mentioned in the previous paragraph, the positive median first order difference spectrograms are used instead of their simpler equivalents used for onset detection. The resulting input vector consists of the spectrograms M_{23}, M_{46}, and M_{92}, as well as the positive median differences D_{23}^{+m}, D_{46}^{+m}, and D_{92}^{+m} for all frames n. The target vector t is built exactly the same way, using the two classes b and $\overline{b}$ (beats and not beats).

3.2.3 Peak detection

For the detection of the beats, the activation function $a_b(n)$ of the unit corresponding to the beat class is used. The processing is performed almost identical to the one of onset detector, with a few minor tweaks outlined in the following.

Thresholding

Since the amount of beats in an audio signals is only a fraction of the onsets, the activation function returned by the neural network differs a bit. The main difference lays in the flatness of the activation function. As a consequence, the median average in the thresholding function θ_o (equation 3.8) is replaced with the mean average, resulting in the threshold θ_b for the beats:

$$\theta_b^* = \lambda_b \cdot \text{mean}\{a_b(0), \ldots, a_b(L)\} \tag{3.20}$$
$$\theta_o = \min\left(\max\left(0.1, \theta_b^*\right), 0.4\right) \tag{3.21}$$

with $a_b(n)$ being the output activation function of the neural network for the beat class b, λ_b a factor determined experimentally on the validation set used for early stopping during the training of the neural network, and L the length of the audio signal. The λ_b^* yielding the best F-measure is chosen. This threshold is then adjusted, to be within certain upper and lower bounds, which evolved during the evaluation process. The resulting beat function $b_b(n)$ containing only activation values $a_b(n)$ within this range and above the calculated threshold θ_b is given by:

$$b_b(n) = \begin{cases} a_b(n) & \text{for } a_b(n) > \theta_b \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

Peak combining

Since the distance between beats are always greater than the human temporal resolution of sounds, a peak combining stage is not necessary for beat detection.

Peak picking

The peak picking stage is exactly the same as for onset detection, resulting in the condition for the beat peak detection being:

$$b(n) = \begin{cases} 1 & \text{for } b_b(n-1) \leq b_b(n) \geq b_b(n+1) \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

3.3 Tempo induction

Based on the beat activation function $a_b(n)$, the tempo can be estimated. The tempo induction method shown in figure 3.3 acts as a drop in replacement for the peak detection stage in the beat detection setup (rightmost rectangle in figure 3.2).

It is structured similar to the peak detection stage of the beat detector. The activation function is thresholded, then the periodicity is detected with the autocorrelation function and finally the highest peak is picked.

Figure 3.3: The tempo induction method with an autocorrelation stage as a replacement for the peak detection stage for beat detection.

Thresholding

The output activations function $a_b(n)$ of the neural network is thresholded with a fixed threshold θ_t value before being processed further. The threshold is determined on basis of the validation set, and is set so that the tempo detection performance gets maximised. During evaluation a value of $\theta_t = 0.075$ showed the best results over a large range of different music and sound examples. The resulting beat activation function for tempo induction $b_t(n)$ is given by:

$$b_t(n) = \max\left(a_b(n), \theta_t\right) \tag{3.24}$$

with $a_b(n)$ being the output activation function of the neural network stage for the beat class b.

Autocorrelation function

Similar to the autocorrelation based tempo induction algorithms (see section 1.2.1), the most dominant interval lengths are used to determine the tempo. As basis for the ACF the thresholded beat function $b_t(n)$ is used, with the resulting function $A(\tau)$ given as:

$$A(\tau) = \sum_n b_t(n+\tau) \cdot b_t(n) \tag{3.25}$$

Considering a given tempo range of the audio signal from T_{min} to T_{max} given in bpm, only values of $A(\tau)$ corresponding to the range of τ_{min} to τ_{max} (given in frames) are used for the calculation. Since the music sightly varies in tempo, and beats are sometimes early or late relative to the absolute position for the dominant tempo, the resulting intervals between beats vary as well. Therefore a smoothing function s is applied to the result of the autocorrelation function $A(\tau)$. A standard hamming window (equation 3.1)

with a size of $\tau_t = 15$ frames is used. The size of this window is not that important, as long as it is wide enough to cover all possible interval fluctuations, and remains shorter than the smallest delay τ_{min} used for the autocorrelation. This results in the smoothed autocorrelation function $A(\tau)$:

$$A^*(\tau) = A(\tau) \star s\,(\tau_t) \tag{3.26}$$

Peak picking

The strongest detected tempo T corresponds to the highest peak in the smoothed autocorrelation function $A^*(\tau)$ at the index τ^*, and can be calculated as:

$$T_{bpm} = \frac{f_f \cdot 60}{\tau^*} \tag{3.27}$$

with f_f denoting the frame rate.

Chapter 4

Data sets and performance measures

This chapter describes the data sets, which were used during the evaluation process and on which the reported results are based. It further lists the measures used for onset, beat, and tempo performance comparison.

4.1 ISMIR 2004 Ballroom set

The data set consists of 698 song excerpts of ballroom dance music with a length of approximately 30 seconds each [57]. Every song has a ground truth tempo which is used for the tempo detection results in the corresponding section. Since all detection methods are purely data driven, other ground truth data (such as the given dance style and hence the meter of the song) is not used for this purpose. The data set is abbreviated BRD from now on. Figure 4.1(a) shows the tempo distribution of the set.

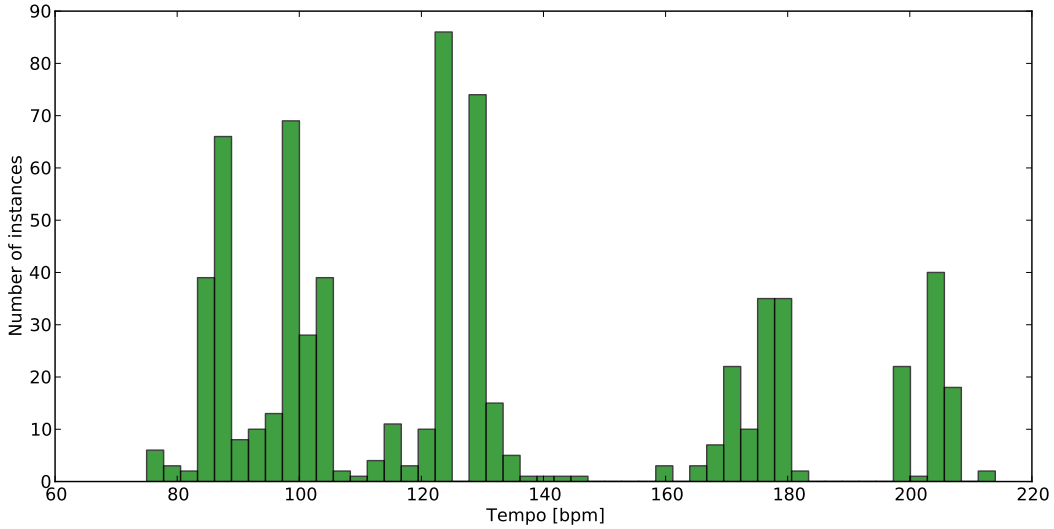
4.2 MTV set

This dataset consists of 200 songs, representing MTV Europe's most wanted top ten of the years 1981 to 2000. Each audio file is of full length and has a manually annotated ground truth tempo. The set is a typical selection of popular music of different genres, such as Pop, Hip Hop, Electronica, Rock, and Ballads. The data set is abbreviated MTV from now on. Figure 4.1(b) shows the tempo distribution of the set.

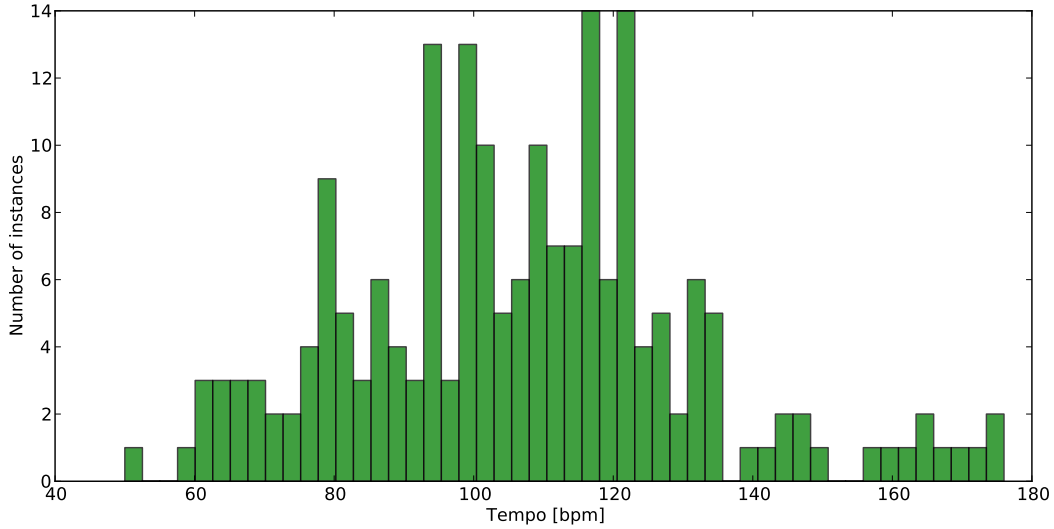
4.3 Juan Pablo Bello set

In [5], Juan Pablo Bello[1] introduced a set consisting of 23 sound excerpts of different lengths with onset ground truth data. It is divided into four categories: pitched per-

[1]Special thanks to Juan Pablo Bello for providing his data set to me!



(a) BRD set



(b) MTV set

Figure 4.1: Tempo distribution of the BRD and MTV sets

cussive (e.g. piano), pitched non-percussive (e.g. bowed strings), non-pitched percussive (e.g. drums), and complex mix (e.g. pop or complex music). The set includes both material generated by synthesisers based on MIDI files as well as audio recordings. The data set is abbreviated JPB from now on, the sub sets PP, NPP, PNP and MIX respectively.

Data set	# files	# orig	# mod	# comb	min / max / mean length [s]
PP	9	489	483	426	2.5 / 60.0 / 10.5
NPP	6	212	222	217	1.4 / 8.3 / 4.3
PNP	1	93	96	93	13.1 / 13.1 / 13.1
MIX	7	271	327	279	2.8 / 15.1 / 8.0

Table 4.1: Number of files and onsets in the JPB data sets, *orig* marking the original onset annotations, *mod* the modified onsets, and *comb* the modified onsets with all onsets within a window of 30 ms combined.

Table 4.1 lists some facts for the data sets. Since all onset annotations were modified during the evaluation process, different numbers of onsets are given. The column marked *orig* corresponds to the original annotations, which are used to compare the onset detection performance of the new approach to some published results. The *mod* column lists the number of onsets for the modified annotations, and *comb* the modified onsets with all onsets within the combination window $c_w = 30$ ms combined to one onset.

To achieve good performance results with the described new neural network approach, highly accurate ground truth data is needed. Thus, all onset annotations other than the MIDI generated ones were modified/corrected for training purpose. Most modifications are a result of matching the annotation style of the manually annotated files as close as possible to the MIDI generated ones. Another large part of the performed modification were due to splitting combined onsets into individual ones. As a consequence, the number of onsets increased considerably in some cases.

The modifications are most visible for the MIX data set. Splitting combined onsets into individual ones and adding missing note changes are the reason for the massive increase of onsets from 271 to 327 onsets. Especially with complex mixes as present in this set, it is not always easy to distinguish individual onsets. So most annotators combine them to a single one. Nonetheless this extra work was done and resulted in 56 extra onsets. Recombining them with a relatively small combination window of only 30 ms (3 frames), the number is reduced to 279 onsets again and thus comparable to the original set, as for all other sets as well.

4.4 Training sets

The training of neural networks with supervised learning requires annotated ground truth material. Therefore the following two sets, containing onset and beat annotations, are introduced.

4.4.1 Onset set

For training of the neural nets the onsets of 110 audio samples were manually annotated. 87 ten seconds excerpts were taken from the BRD data set, the remaining 23 from the already annotated JPB data set. Part of the annotation work was already done by Alexandre Lacoste and Douglas Eck for training their neural net approach [61][2]. Nonetheless, as with the JPB data set, all markers were corrected and missing onsets were added to match the annotation style of the other files and the precision of the MIDI based files. The complete set consists of 6605 onsets. If onsets within a combination window of 30 ms are unified, the number decreases to 5861.

4.4.2 Beat set

For the beat and tempo detection task, beat ground truth data is needed. Thus 91 out of the before mentioned 110 audio samples were also beat annotated. The discrepancy in the number is a result of the fact that some samples do not consist of music with a beat, but are rather of synthetic type (e.g. bells), and could therefore not be annotated. For the sake of simplicity, only the already onset annotated audio samples have been marked with additional beat positions. The whole set consists of 2053 beats.

4.5 Performance measures

For evaluating the performance the following measurements are introduced. Precision-rate (P), recall-rate (R), and F-measure (F) are calculated as follows:

$$P = \frac{C}{C + F^+} \tag{4.1}$$

$$R = \frac{C}{C + F^-} \tag{4.2}$$

$$F = \frac{2PR}{P + R} \tag{4.3}$$

[2]The annotations can be downloaded at `http://w3.ift.ulaval.ca/~allac88/dataset.tar.gz`

with C being the number of correctly identified onsets or beats, F^+ the number of false positives, and F^- the number of false negatives. Those measures are used for the evaluation of the onset and beat detection performance. The F-measure is a single value that gives the overall performance of the best balance between the precision and recall rates.

For printing receiver operating characteristic (ROC) curves, the percentages of true positives (TP) and false positives ($FP1$) are needed and calculated as:

$$TP = \frac{C}{C + F^-} \tag{4.4}$$

$$FP = \frac{F^+}{C + F^+} \tag{4.5}$$

An onset is considered as correctly identified, if it lays within a certain detection window around the annotated ground truth onset. If not stated otherwise, a detection window of 50 ms is used for onset detection. It consists of the annotated frame of 10 ms width plus two frames of 20 ms on each side of the onset. If compared to other results, the same detection window as for the original results is used. For beats, the detection window is widened to 70 ms.

For measuring the tempo induction performance, two accuracies are used. Accuracy 1 gives the percentage of songs with correctly identified tempo, while accuracy 2 also includes the octave errors.

$$\text{Accuracy 1} = \frac{C_{nominal}}{N} \tag{4.6}$$

$$\text{Accuracy 2} = \frac{C_{octaves}}{N} \tag{4.7}$$

with N the total number of instances, $C_{nominal}$ the number of instances with correctly identified tempo, and $C_{octaves}$ the number of correctly identified tempo, if also doubles, halves, triples, and thirds of the ground truth tempo are included. A tempo is considered as correct, if it deviates less than 4% from the annotated ground truth tempo, like in [44].

Chapter 5

Evaluation

This chapter describes all performed tests and evaluations to get the final onset, beat, and tempo detection system described in chapter 3. In addition, the used parameters for the final tests and the results reported in chapter 6, were determined during this process.

Throughout all evaluations, the frame rate was set to 100 frames per seconds. This gives both a good resolution of 10 ms for one frame, and keeps the computational complexity at a reasonable level. A smaller frame rate would be unacceptable with regard to tempo induction. An increased frame rate of e.g. 200 fps would not only double all calculations, it would also require the annotations being more accurate, but an accuracy of 5 ms for the annotations is almost impossible to achieve, thus sticking with 100 fps.

5.1 Onset detection

For the new onset detector, the input representation for the neural network, the type and topology of the neural network, and the onset classifier had to be evaluated. This process is covered in the following subsections.

5.1.1 Input representation

Finding a good input representation for an artificial neural network is the first step. As expected, test runs with time domain representations of the signal didn't show acceptable results, hence only tests within the frequency domain are evaluated.

Linear spectrogram

Tests with the complete STFT spectrogram with a standard window length of 23.22 ms (1024 frames) gave promising results. Training neural networks with input vectors with

such a high dimensionality takes a very long time. Whatever neural net topology is chosen (see later in this section), the most connections result from connecting the inputs to the first hidden layer. So the most effective solution to lower the needed training time is to reduce the input vector size.

Mel spectrogram

A standard way of reducing the frequency domain representation of an audio signal without affecting the performance [69] is to transform the data to the Mel scale. The Mel scale is a perceptual scale of the pitch of a tone and has a linear frequency response below 1127 Hz and a logarithmic scale above this frequency. Although the human ear has a resolution of 24 critical bands [32], test showed that using higher number of bands gave slightly better results.

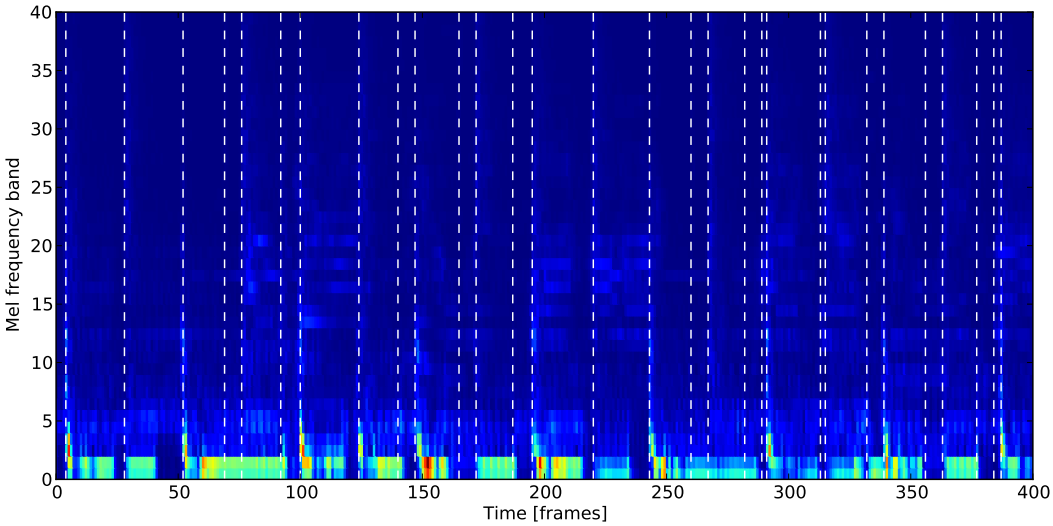
Using 40 Mel bands gives a good compromise between acceptable training times for the neural network and a performance still on par with the original STFT spectrogram. An explanation for the observation that this transformation to the Mel scale works without any major performance penalty, is that the computation of each Mel band in the frequency domain is a simple weighted summation of the STFT spectrogram's frequency bins. Since each unit in a neural network acts as a simple summariser and adds up all inputs with a certain weighting applied, this summation can be either done by the neural network itself or as a preprocessing step with the input signals frequency domain representation.

Logarithmic Mel spectrogram

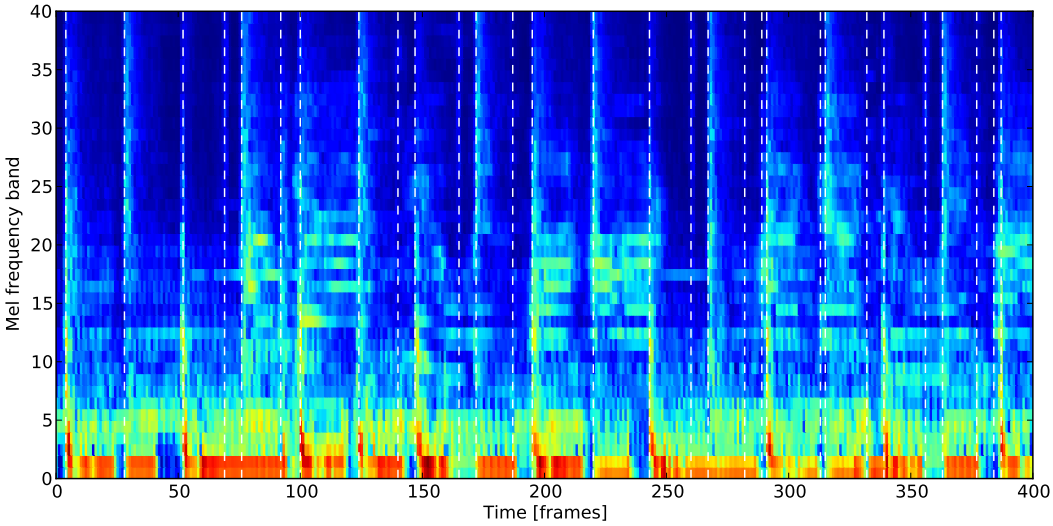
The human ear does not only have a nonlinear perception of frequencies, but also a nonlinear perception of loudness. Taking this into account by using the logarithmic magnitudes instead of the linear ones tweaked the results considerably. The visualisation of this measure is illustrated in figure 5.1. Especially the higher frequencies with much lower magnitudes are more highlighted afterwards.

STFT window sizes

So far only one spectrogram with a window size of 23.22 ms was considered. Inspection of the missed onsets showed that the system mostly failed at note changes. Visual checks of the spectrogram revealed that some onsets are not clearly visible because of the low frequency resolution with this STFT window length. This brought up the idea of including a spectrogram with a better frequency resolution as well. Window lengths



(a) Linear Mel spectrogram



(b) Logarithmic Mel spectrogram

Figure 5.1: The linear Mel spectrogram and its logarithmic counterpart for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*. A STFT window of 23.22 ms was used, onsets are marked with dashed lines.

of 46.44 and 92.88 ms (2048 and 4096 frames) were tested. Figure 5.2 shows how the frequency resolution increases with bigger STFT window sizes, mostly visible in the low frequency bands, where a change in tones occurs around frame numbers 220 and 330. On the contrary, the exact location of these changes can be better extracted from the spectrograms with shorter STFT windows. Thus the potential of a shorter window length of 11.61 ms (512 frames) to improve the precision of the onsets was investigated, too.

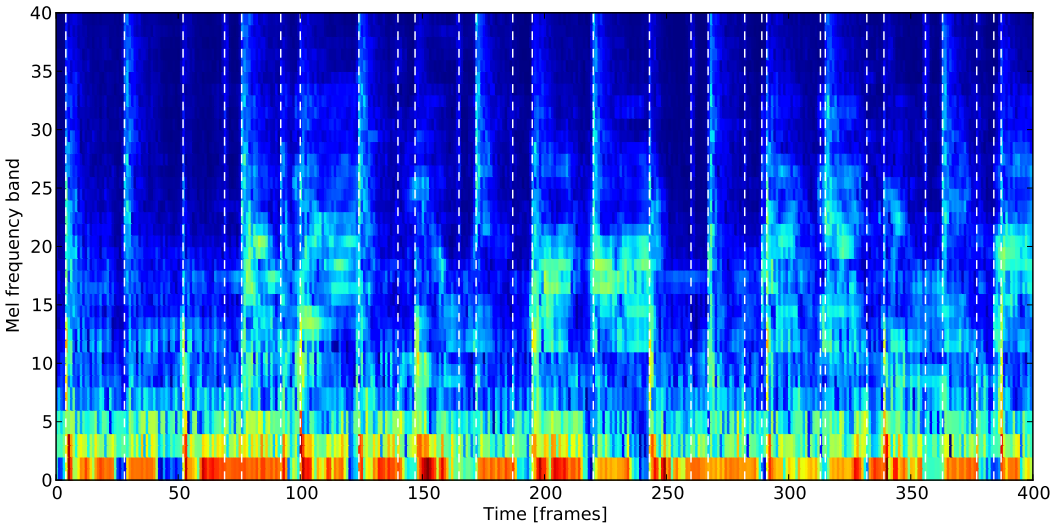
Table 5.1 shows the performance of spectrograms with different window sizes and some combinations thereof. It can be seen that the combination of different spectrograms always had a positive effect.

Input	Precision	Recall	F-measure	TP[%]	FP[%]
M_{12}	0.837	0.829	0.833	82.9	16.3
M_{23}	0.856	0.855	0.856	85.5	14.4
M_{46}	0.846	0.845	0.846	84.5	15.4
M_{12} M_{23}	0.861	0.860	0.860	86.0	13.9
M_{23} M_{46}	0.881	0.878	0.880	87.8	11.9
M_{12} M_{23} M_{46}	0.885	0.885	**0.885**	88.5	11.5

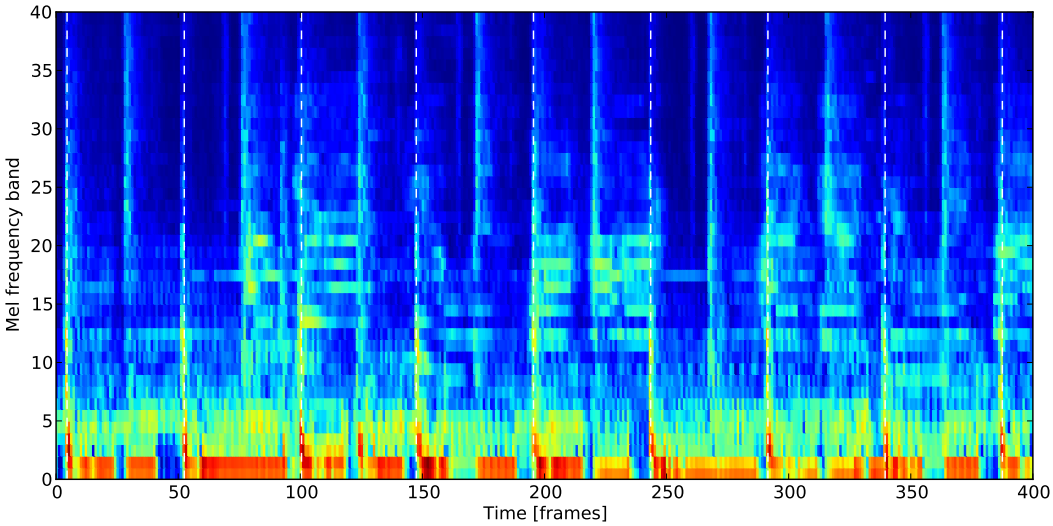
Table 5.1: Onset detection performance of different input representations for a BLSTM network with three hidden layers and 20 units each. Given are the precision, recall, and F-measure rates as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the onset set.

First order differences

Well performing signal feature based onset detection methods often take the change or rise in energy as a feature. Thus, in addition to the normal spectrogram the inclusion of the first order difference and the positive first order differences were investigated as well. Both additions had a positive effect, but the positive difference always performed better than the normal first order difference, as shown in table 5.2. The table gives an overview of the achieved performances when combining different STFT window sizes and positive first order differences of the spectrograms. Although the inclusion of the spectrogram with a window length of 11.61 ms (512 frames) in addition to the 23.33 and 46.44 ms (1024 and 2048 frames) spectrograms had a positive effect (last line of table 5.1), this effect was levelled out when the positive differences of the spectrograms were included as well. In order to keep the input vector size for the neural network as small as possible,

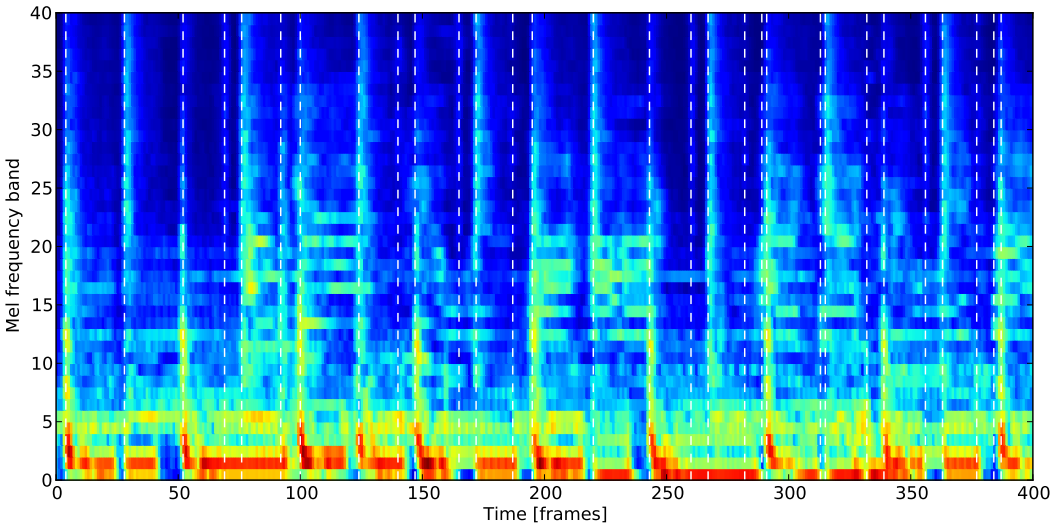


(a) STFT window: 11.61 ms, onsets marked with dashed lines.

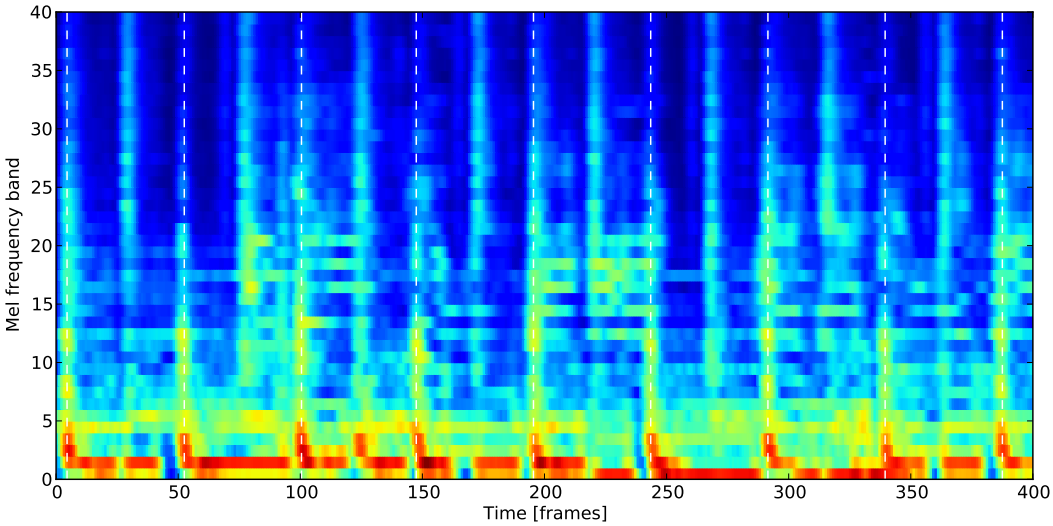


(b) STFT window: 23.22 ms, beats marked with dashed lines.

Figure 5.2: Logarithmic Mel spectrograms with different STFT window sizes for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*.



(c) STFT window: 46.44 ms, onsets marked with dashed lines.



(d) STFT window: 92.88 ms, beats marked with dashed lines.

Figure 5.2: 2 Logarithmic Mel spectrograms with different STFT window sizes for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*. (cont)

only the spectrograms with an STFT window lengths of 23.33 and 46.44 ms and their positive first order differences were chosen.

Input	Precision	Recall	F-measure	TP[%]	FP[%]
$M_{12}\ M_{23}\ D_{12}\ D_{23}$	0.869	0.868	0.868	86.8	13.1
$M_{12}\ M_{23}\ D^+_{12}\ D^+_{23}$	0.880	0.877	0.878	87.7	12.0
$M_{23}\ M_{46}\ D_{23}\ D_{46}$	0.887	0.886	0.886	88.6	11.3
$M_{23}\ M_{46}\ D^+_{23}\ D^+_{46}$	0.900	0.900	**0.900**	90.0	10.0
$M_{12}\ M_{23}\ M_{46}\ D_{12}\ D_{23}\ D_{46}$	0.890	0.889	0.889	88.9	11.0
$M_{12}\ M_{23}\ M_{46}\ D^+_{12}\ D^+_{23}\ D^+_{46}$	0.900	0.900	0.900	90.0	10.0

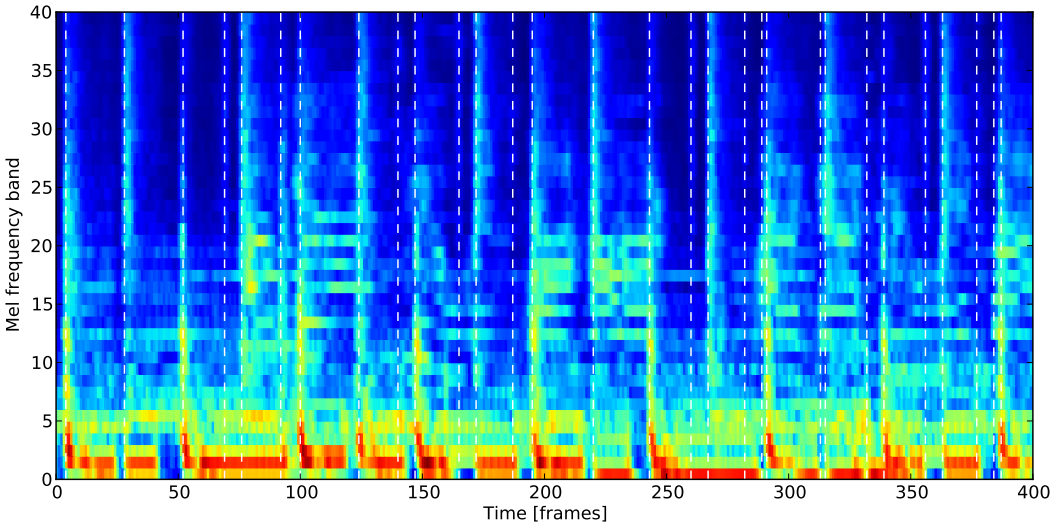
Table 5.2: Onset detection performance of different input representations for a BLSTM network with two hidden layers and 20 units each. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the onset set.

Figure 5.3 depicts the positive first order difference of the logarithmic Mel spectrogram and the underlaying spectrogram itself. Since the difference represents the onset locations almost perfectly, an attempt to further reduce the size of the input vector by only taking the positive first order difference and omitting the original spectrogram was made. However, this gave inferior results. This suggests that not only the difference, but also the actual amount of energy is important for onset detection.

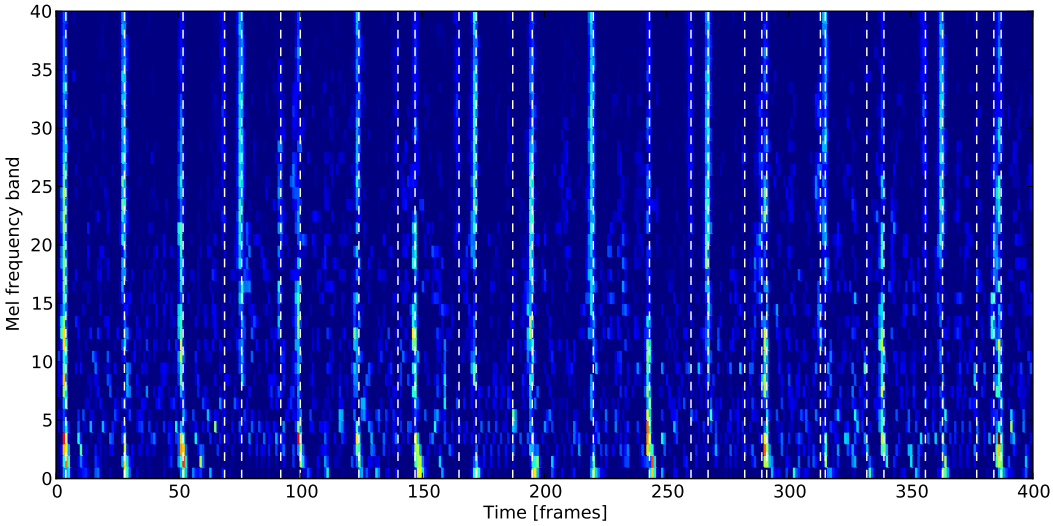
Although the Mel transformation of the spectrograms gives a massive reduction of the input vector size, the resulting input vector is still big if more window lengths and their positive first order differences are used. Including the positive first order differences doubles the input vector size and thus raises the training time of one epoch by a factor of 1.75. But it also entails a reduction in the needed training epochs from 70 to 54 on average. So the overall computation penalty is only a factor of 1.35.

5.1.2 Input normalisation

During evaluation, no performance gain could be achieved by normalising the used input representations, therefore this step is skipped. This observation is consistent with the fact that the inclusion of the positive first order difference gave better results than the inclusion of the normal first order difference (which includes both positive and negative values).



(a) Logarithmic Mel spectrogram



(b) Positive first order difference of the logarithmic Mel spectrogram

Figure 5.3: Logarithmic Mel spectrogram and the positive first order difference of a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*. A STFT window size of 46.44 ms was used, onsets are marked with dashed lines.

5.1.3 Network type

Having found a good working input representation for the neural network, the next step is to explore a good working network type. Although research has shown that BLSTM networks should perform best [49, 46, 47, 48], a cross check was performed to confirm this assumption. Table 5.3 details the results and also lists the average number of epochs needed for training.

It can be seen that bidirectional network types perform better than their unidirectional counterparts. Since bidirectional networks have access to information before and after an onset, they could detect the onsets more reliably. An additional performance gain could be achieved by using Long Short-Term Memory units instead of conventional logistic ones. Both gains are pretty small and are only of significance if both measures are combined. The small advantage of the Long Short-Term Memory based type compared to the standard recurrent neural network type suggests that the information needed for onset detection is located in the direct neighbourhood of the onset itself, thus not completely revealing the potential of Long Short-Term Memory units with their access to temporal distant information.

Network type	Precision	Recall	F-measure	TP[%]	FP[%]	# epochs
RNN	0.884	0.890	0.887	89.0	11.7	110
BRNN	0.886	0.892	0.889	89.2	11.5	100
LSTM	0.890	0.888	0.889	88.8	11.0	60
BLSTM	0.899	0.893	**0.896**	89.3	10.1	50

Table 5.3: Onset detection performance of different neural network types. All networks have two hidden layers with 20 units each. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP), and the number of required training epochs for the validation set of a 8-fold cross validation run on the onset set.

5.1.4 Network topology

Although there exist a number of "rules of thumb" for choosing the right network size and layout, none of them resulted in a good performing network. So the most basic strategy of starting with only one hidden layer with a small number of units and then increasing both parameters gradually was used to determine the best network topology.

Networks with only one hidden layer perform better the more hidden units are used. However the performance saturates after a certain size is reached (upper section of

table 5.4). Superior results can be achieved if more hidden layers are used. Networks with two hidden layers perform considerably better than networks with only one hidden layer, independently of the number of hidden units. The middle section of table 5.4 shows only topologies with units equally distributed since these excel all topologies with unequally distributed hidden units. Contrary to networks with a single hidden layer, increasing the number of units does not lead to better performance automatically. Twenty units per layer emerged as the best size for the two layer topology. Adding a third layer with the same size gives slightly better results (but only the case of twenty units is given in the last line of table 5.4), so this topology with three hidden layers with twenty units each is used for all future onset detection tests.

Topology	Precision	Recall	F-measure	TP[%]	FP[%]
1 layer, 10 units	0.863	0.862	0.863	86.2	13.7
1 layer, 15 units	0.866	0.866	0.866	86.6	13.4
1 layer, 20 units	0.869	0.869	0.869	86.9	13.1
1 layer, 25 units	0.872	0.871	0.871	87.1	12.8
1 layer, 30 units	0.872	0.871	0.872	87.1	12.8
2 layers, 10 units each	0.886	0.885	0.886	88.5	11.4
2 layers, 15 units each	0.895	0.894	0.895	89.4	10.5
2 layers, 20 units each	0.896	0.896	0.896	89.6	10.4
2 layers, 25 units each	0.893	0.893	0.893	89.3	10.7
2 layers, 30 units each	0.889	0.888	0.888	88.8	11.1
3 layers, 20 units each	0.900	0.899	**0.900**	89.9	10.0

Table 5.4: Onset detection performance of different BLSTM topologies with varying numbers of used hidden layers and units per hidden layer. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the onset set.

5.1.5 Network training and testing

For training, the neural network needs the input information and the corresponding target information. For the onset detection method the classification task is chosen. Therefore the signal is processed to be present in the above mentioned representation and the onset annotations are used as the target information. Each labelled onset is mapped to the onset class o, and all other positions to the second non-onset class $\bar{o}$.

All weights are initialised randomly with a mean of 0 and a standard deviation of 0.1. As the learning algorithm, steepest descent with a momentum of 0.9 and a learn rate

of 0.0001 is used throughout all evaluation tests. Whenever different network related parameters are changed (e.g. network type or topology), the weights are initialised with the same random seed, to guarantee the comparability of the results.

If parameters are to be determined on the basis of the validation set, a minimum of ten complete training runs with randomised seed were performed. This is necessary to level out the influence of the randomly chosen initial weights.

Network training is performed with the early stopping method, thus the complete data set has to be split into three non overlapping sets, namely the training set, the validation set, and the testing set. For network training, 8-fold cross validation is used. The complete annotated set of 110 files was therefore randomly split into eight sub sets. For each run, six sub sets were used for training and one for validating and one for testing respectively. The use of each sub set was changed between different runs, so after eight runs each sub set was used once for validating and testing the remaining sets. All validation and test results for the files are then combined to get the complete validation and test sets.

The standard classifier for neural networks works simply by choosing the output unit with the highest activation as the winner and classifies the input accordingly. This method is also used for training purposes when back propagating the error and adjusting the weights. For the final classification into the two classes o and $\bar{o}$, it is not usable as it is, and thus a special classification algorithm needs to be used.

5.1.6 Onset classification

For the onset detection a classification method must be developed, which works with the standard output activations produced by the neural network, but also detects the onsets with activation values below the standard threshold of 0.5 (for the case of two used classes). Figure 5.4 shows the output activation function of the onset class unit (blue line) together with the onset targets marked as vertical green dotted lines.

A very simple approach of thresholding was chosen. A threshold is calculated for each file, depending on the median average of the onset activation function. This threshold can be seen in figure 5.4 as the horizontal blue dotted line. All local maxima above this threshold are considered as onsets. An onset is identified correctly if it lays within the detection window of 50 ms (the annotated frame of 10 ms width plus two frames of 20 ms on each side of the onset). For the final results, this detection window was set to different values, depending on the size of the window that was used for the results which the new approach is compared to.

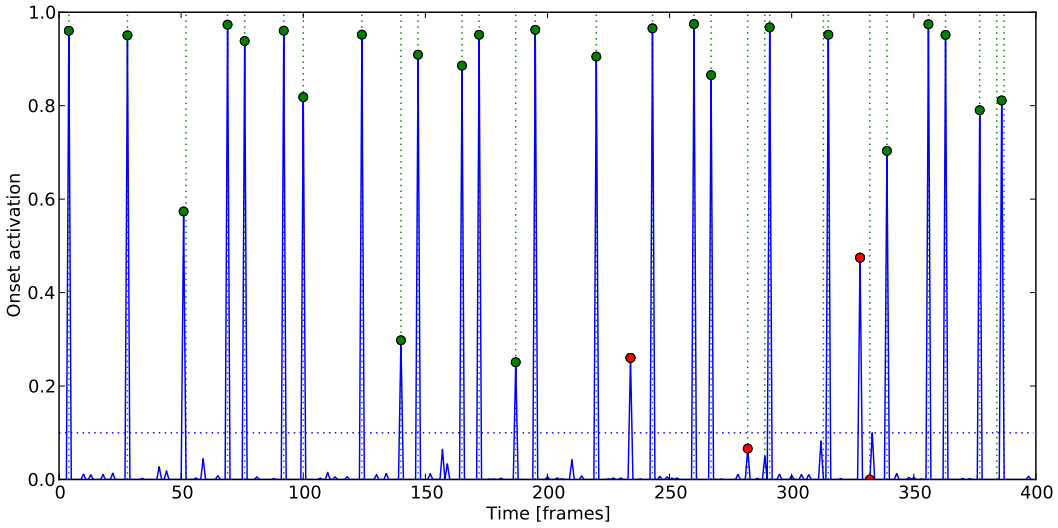


Figure 5.4: Output activation function for the onset class o of a BLSTM network with 3 hidden layers with 20 units each (blue), ground truth onsets (green), correctly identified onsets (green dots), missed or false onsets (red dots), and a fixed threshold $\theta_o = 0.1$ (blue dotted) for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*.

For the calculation of the threshold different approaches have been examined. The method of a hard threshold, the mean and median averaging algorithms, and the standard deviation. They all showed similar performances, with a small advantage for the averaging algorithms. The decision towards the median algorithm was made, because it showed the best overall performance over all data sets. For all variants a multiplication factor (λ_o in equation 3.8) was chosen such that the F-measure gets maximised. During evaluation, a factor of 50 gave very good results for all kind of onset and music types.

One action all algorithms profited from, was the limitation of the threshold to a lower and an upper bound. In a sense, a fixed threshold is noting more, than setting both limits to the same value. During extensive tests, the best values emerged as 0.1 and 0.3 and are used in equation 3.9.

5.2 Beat detection

The beat detection process is almost the same as the one described for onset detection in the previous section. Thus, this section is outlined very similar to the previous one.

5.2.1 Input representation

As a starting point, the same input representation as for the onset detection was used. Since it is always desirable to keep the input vector as small as possible, it was investigated whether the number of Mel bands could be further reduced. It turned out that for beat detection purposes twenty bands are enough.

First order differences

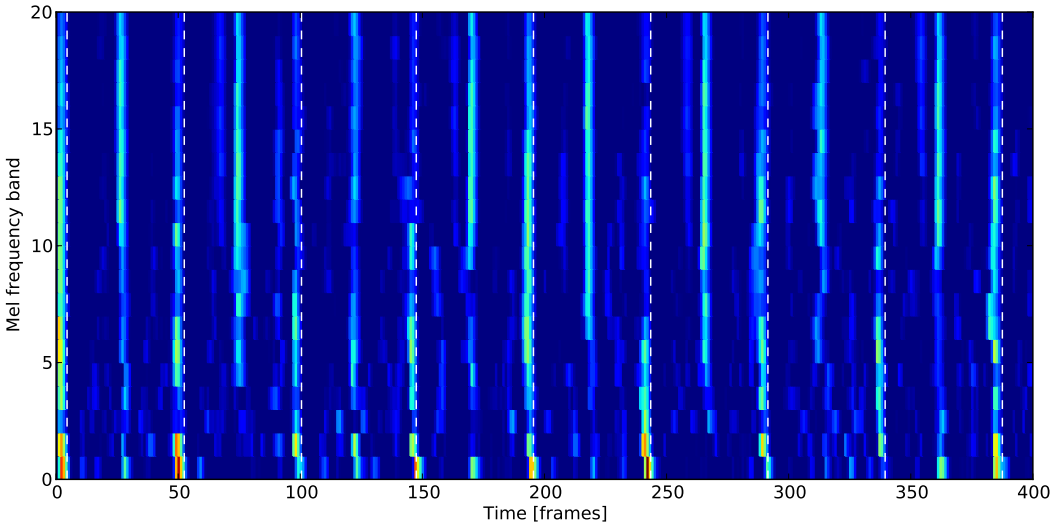
Identically to the onset detection, the inclusion of the positive first order difference always had a positive effect. However, compared to the onset detection, the overall achieved detection rates are lower. Visual inspection of the false negative beats and the corresponding spectrograms revealed that especially in pitched non-percussive music, beats often occur at the same time as long held tones. Thus alternative first order difference calculation methods, which take longer ranges into account, were investigated.

First attempts with building a moving average over the last ten frames showed good results for the STFT window size of 1024 frames. However, this effect was less visible if longer STFT windows were used. Increasing the range over which the average was built regained the performance boost for longer window sizes. Thus the average is calculated over a range dependent on the window size used. The first order difference is then calculated against this moving average. Table 5.5 lists the different used averaging algorithms and the sizes over which the averages were built. A quite impressive additional performance boost of three percent points can be achieved, if the median with a window size equal to one hundredth of the STFT window size is used instead of the simple positive first order difference. For the exact calculation, please refer to section 3.2.1.

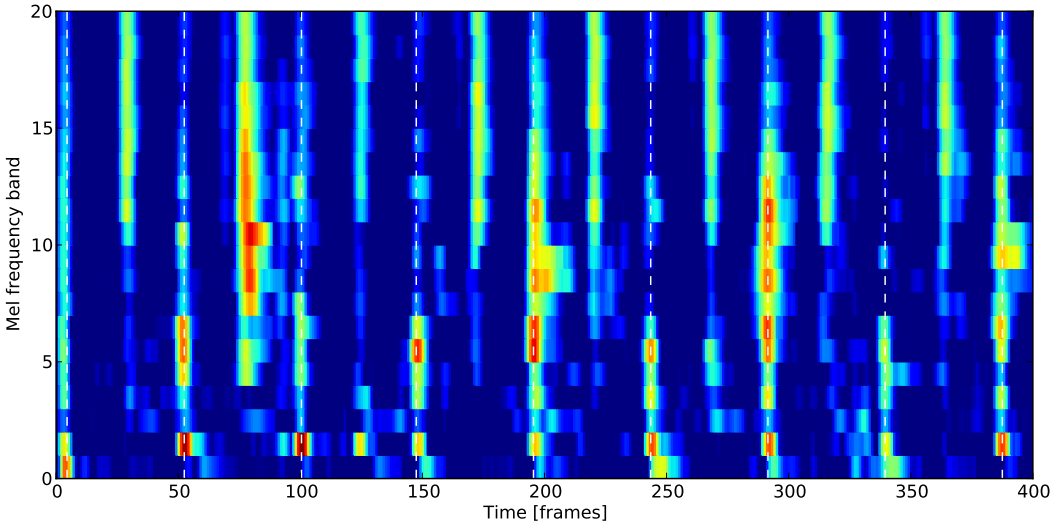
Figure 5.5 shows the positive first order differences to the preceding frame (top) and to the median over the last 0.41 s (bottom) for the spectrogram with a STFT window of 4096 frames. A possible explanation that the median version performs that much better might be that the median version not only considers a wider window, but also that this measure minimises the displacement of the peak values from the actual beat positions.

STFT window sizes

Again, different sizes for the STFT window were analysed. Since the median difference, which considers longer ranges, performed better than the simple first order difference, a similar gain was expected when including longer STFT window sizes. Additionally lengths of 11.61, 92.88 and 185.76 ms (512, 4096 and 8192 frames) were investigated. The shortest window length of 11.61 ms performed worst, and always had a negative



(a) Positive first order difference to the preceding frame



(b) Positive first order difference to the median of the last 0.41 s (41 frames)

Figure 5.5: Visualisation of different first order difference representations of the logarithmic Mel spectrogram for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu* with a STFT window size of 46.44 ms. Beats are marked with dashed lines.

Input	Precision	Recall	F-measure	TP[%]	FP[%]
Δ^+_{n-1}	0.773	0.773	0.773	77.3	22.7
$\Delta^+_{mean\{n-N/50,\ldots,n-1\}}$	0.799	0.796	0.798	79.6	20.1
$\Delta^+_{mean\{n-N/100,\ldots,n-1\}}$	0.800	0.800	0.800	80.1	20.0
$\Delta^+_{mean\{n-N/200,\ldots,n-1\}}$	0.793	0.793	0.793	79.3	20.7
$\Delta^+_{median\{n-N/50,\ldots,n-1\}}$	0.800	0.801	0.800	80.1	20.0
$\Delta^+_{median\{n-N/100,\ldots,n-1\}}$	0.810	0.807	0.809	**80.7**	19.0
$\Delta^+_{median\{n-N/200,\ldots,n-1\}}$	0.794	0.798	0.796	79.8	20.8

Table 5.5: Beat detection performance of different input representations for a BLSTM network with three hidden layers with 25 units each. As inputs the spectrograms with the window sizes of 23.22 and 46.44 ms and different first order difference spectrograms with the same window sizes are used. Δ^+_{n-1} denotes the positive first oder difference to the preceding frame, Δ^+_{mean} the positive difference to the mean (Δ^+_{median} to the median) average of the given range, with n being the frame index and N the used STFT window size in frames. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the beat set.

impact if combined with other window lengths. Mixtures of different sizes greater than 23.22 ms gave better results than used alone. The combination of the three window lengths 23.22, 46.44, and 92.88 ms performed best, better than all combinations of two lengths. Given the fact that the positive median first order difference always gave the best results, table 5.6 only shows results for different window lengths of the spectrograms obtained with the inclusion of that first order difference.

Onset information

[41] made an evaluation of low level features for beat tracking and concluded that the best feature sets are those including features that indicate onsets. [84] investigated the relation of beats and onsets positions and observed that beats occur at onset positions almost all of the time. Thus different forms of onset information were incorporated into the input vector: the detected onsets, the raw onset activation values, and for comparison the ground truth onset data. All methods degraded the performance, so this measure was rolled back. A possible explanation is that the neural network gives these values too much weight and therefore produces much more false positives. Table 5.7 shows that this effect is more pronounced for less accurate onset information, ground truth onset data degraded the performance lesser than the other additional informations.

Input	Precision	Recall	F-measure	TP[%]	FP[%]
M_{12} D_{12}^{+m}	0.749	0.749	0.749	74.9	25.1
M_{23} D_{23}^{+m}	0.792	0.789	0.790	78.9	20.8
M_{46} D_{46}^{+m}	0.804	0.804	0.804	80.4	19.6
M_{93} D_{93}^{+m}	0.785	0.782	0.783	78.2	21.5
M_{186} D_{186}^{+m}	0.784	0.778	0.781	77.8	21.6
M_{12} M_{23} D_{12}^{+m} D_{23}^{+m}	0.774	0.772	0.773	77.2	22.6
M_{23} M_{46} D_{23}^{+m} D_{46}^{+m}	0.810	0.807	0.809	80.7	19.0
M_{46} M_{4069} D_{46}^{+m} D_{93}^{+m}	0.803	0.800	0.801	80.0	19.7
M_{4069} M_{186} D_{93}^{+m} D_{186}^{+m}	0.790	0.793	0.791	79.3	21.0
M_{12} M_{23} M_{46} D_{12}^{+m} D_{23}^{+m} D_{46}^{+m}	0.800	0.800	0.800	80.0	20.0
M_{23} M_{46} M_{93} D_{23}^{+m} D_{46}^{+m} D_{93}^{+m}	0.815	0.815	**0.815**	81.5	18.5
M_{46} M_{93} M_{186} D_{46}^{+m} D_{93}^{+m} D_{186}^{+m}	0.811	0.811	0.811	81.1	18.9

Table 5.6: Beat detection performance of different input representations for a BLSTM network with three hidden layers with 25 units each. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the beat set.

Input	Precision	Recall	F-measure	TP[%]	FP[%]
w/o onset information	0.815	0.815	**0.815**	81.5	18.5
with ground truth onsets	0.813	0.814	0.813	81.4	18.7
with detected onsets	0.801	0.799	0.800	79.9	19.9
with onset activations	0.792	0.791	0.792	79.1	20.8

Table 5.7: Beat detection performance with different types of onset information in addition to the Mel spectrograms and positive median first order differences with window sizes of 23.22, 46.44, and 92.88 ms for a BLSTM network with three hidden layers with 25 units each. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the beat set.

5.2.2 Input normalisation

Again, no performance gain could be achieved by normalising the used input representations. This was expected, because of the observations made for the onset detection method (see section 5.1.2).

5.2.3 Network type

For onset detection, both the bidirectional recurrent network and the unidirectional Long Short-Term Memory network perform better than the standard logistic unidirectional one, but give only a negligible amount of extra performance. The combination of Long Short-Term Memory units and the bidirectional structure gives at least a measurable performance boost.

For beat detection, both the use of bidirectional networks and Long Short-Term Memory units improve the results measurably. If these two measures are combined, the resulting bidirectional Long Short-Term Memory network reveals its real capabilities, with a considerably better performance than all other network types (see table 5.8). This suggests that the detection of beats depends much more on distant information in both temporal directions than onsets. Also the needed epochs for training are reduced by a much larger amount compared to onset detection.

Network type	Precision	Recall	F-measure	TP[%]	FP[%]	# epochs
RNN	0.662	0.655	0.658	65.5	33.8	330
BRNN	0.700	0.705	0.703	70.5	30.0	300
LSTM	0.697	0.697	0.697	69.7	31.3	102
BLSTM	0.815	0.815	**0.815**	81.5	18.5	65

Table 5.8: Beat detection performance of different neural network types. All networks have three hidden layers with 25 units each. Given are the precision, recall, and F-measure as well as the percentage of true (TP) and false positives (FP), and the number of required training epochs for the validation set of a 8-fold cross validation run on the beat set.

5.2.4 Network topology

Once again the aforementioned strategy of starting with only one hidden layer with a small number of units and then increasing both parameters gradually was used to determine the best network topology.

Interestingly, this time the behaviour of the networks seem more predictable, as adding more layers improves the performance, and the optimal number of units per layer remains the same, independent of the number of used layers. Again, if more layers are used, the networks with evenly distributed units perform better than ones with unevenly distributed units, thus table 5.9 only shows those topologies. The last line shows the winning network, consisting of three layers with 25 LSTM units each.

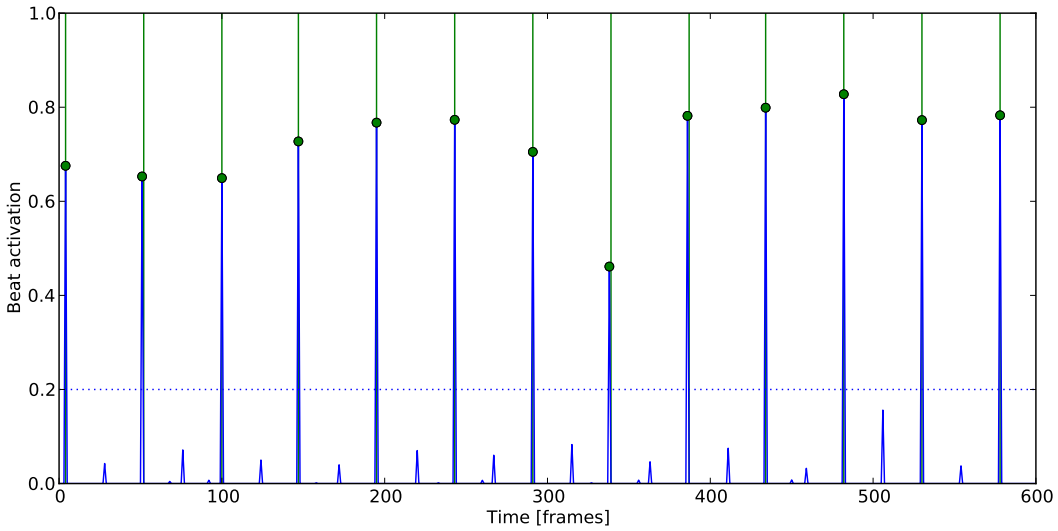
Network topology	Precision	Recall	F-measure	TP[%]	FP[%]
1 layer, 10 units	0.771	0.770	0.770	77.0	22.9
1 layer, 15 units	0.783	0.783	0.783	78.3	21.7
1 layer, 20 units	0.788	0.787	0.788	78.7	21.2
1 layer, 25 units	0.790	0.790	0.790	79.0	21.0
1 layer, 30 units	0.777	0.777	0.777	77.7	22.3
2 layers, 10 units each	0.780	0.779	0.779	77.9	22.0
2 layers, 15 units each	0.789	0.789	0.789	78.9	21.1
2 layers, 20 units each	0.791	0.792	0.792	79.2	20.9
2 layers, 25 units each	0.793	0.794	0.793	79.4	20.0
2 layers, 30 units each	0.783	0.783	0.783	78.3	21.7
3 layers, 25 units each	0.815	**0.815**	0.815	81.5	18.5

Table 5.9: Beat detection performance of different topologies of BLSTM networks with varying numbers of used hidden layers and units per hidden layer. Given are the precision, recall, and F-measure as well as the percentage of true (TP) and false positives (FP) for the validation set of a 8-fold cross validation run on the beat set.

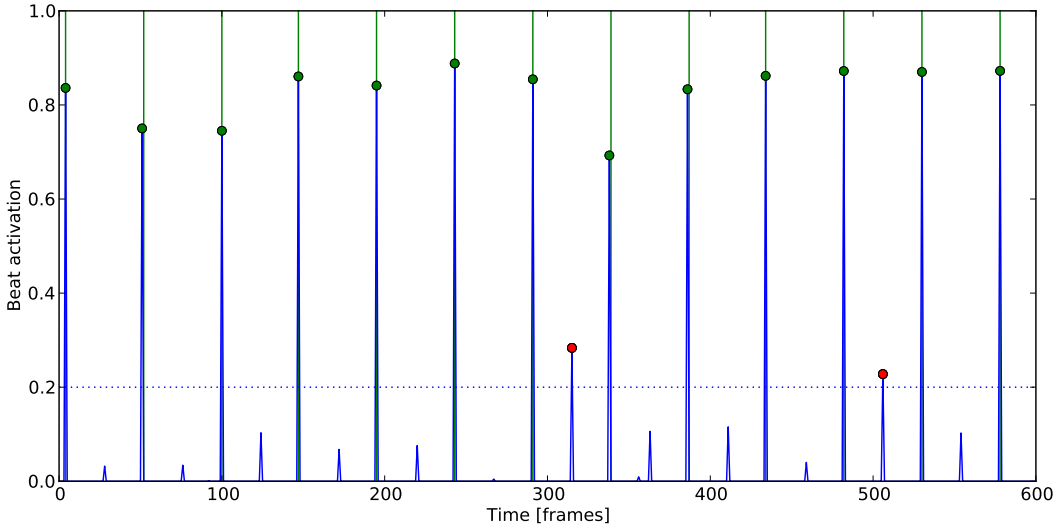
5.2.5 Network training and testing

The network training and testing procedure is the same as for onset detection, and is described in section 5.1.5. The only difference is that the 87 beat annotated files are used instead of the 110 from the onset data set.

Since the beat set only contains 2053 annotated beats, the beat detection performance depends largely on how the data set is split into the training, validation and test sets. Figure 5.6 illustrates how the detection of beats can change if the set is split differently. This behaviour was never experienced during onset training. As a side note, the images also perfectly illustrate the often experienced case, where the double tempo is inducted instead of the correct one.



(a) 8-fold cross validation configuration I



(b) 8-fold cross validation configuration II

Figure 5.6: Different output activation functions (blue) for the beat class b of a BLSTM network with 3 hidden layers with 25 units each, beat ground truth data (green), correctly identified beats (green dots), false beats (red dots), and a fixed threshold $\theta_b = 0.2$ (blue dotted) for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu* with altered partitioning of the beat set into training and validation sets.

5.2.6 Beat classification

Since the general structure of the beat detector is almost the same as the one for onsets, only little modifications to the classification algorithm described in section 5.1.6 were made.

The amount of beats in an audio signal is only a fraction of the onsets, so the median average of the beat activation function is closer to zero (see figure 5.6(a)) and thus not usable for the calculation of the threshold. It is replaced with the mean average, which gives more reliable results. The multiplication factor (λ_b in equation 3.20) was chosen such that the F-measure gets maximised. During evaluation, a factor of 8 gave very good results for all kind of beats. Also the minimum and maximum thresholds are modified and set to 0.1 and 0.4 respectively. A beat is considered as identified correctly if it is detected within a window of 90 ms (the annotated frame of 10 ms width plus 4 frames or 40 ms on each side of the beat).

The peak combining step is not needed for beat detection, since beats are always located farther apart from each other. Theoretically, the neural network can detect beats close together as well, but this was never observed during evaluation, thus this step is omitted.

5.3 Tempo induction

As described in section 3.3, the tempo induction depends only on a few parameters which must be determined. The first is the threshold θ_t for the beat activation function. Table 5.10 shows that it is advantageous to use a certain threshold for the beat activation function returned by the neural net, before the autocorrelation function is used to detect the dominant inter beat intervals.

It can be observed that higher thresholds θ_t lead to degraded performance. This complies with the results in [23, 44], where histogram based methods perform worse than autocorrelation based methods. The higher the threshold is set, the more the ACF method practically transforms into a histogram based method.

The second variable is the range of tempo which is considered for the autocorrelation function. Different lower and upper bounds were inspected. The results shown in table 5.11, reveal that a lower limit of 70 bpm and an upper limit of 225 bpm performed best. This tempo range is exactly the maximum tempo range of the data set (the distribution is depicted in figure 4.1(a)) plus the four percent accuracy tolerance of the performance measure.

BRD [%]	Accuracy 1	Accuracy 2
θ_t=0.00	72.6	94.3
θ_t=0.05	73.5	**95.3**
θ_t=0.75	**75.2**	95.1
θ_t=0.10	74.5	94.4
θ_t=0.15	73.5	94.1

Table 5.10: Tempo induction performance of different beat activation function thresholds for the autocorrelation function of the tempo induction algorithm. As inputs for the BLSTM network with three hidden layers with 25 units each, Mel spectrograms and positive median first order differences with window sizes of 23.22, 46.44, and 92.88 ms are used. Given are the Accuracies 1 and 2 for the BRD set.

BRD [%]	Accuracy 1	Accuracy 2
60 . . . 210 bpm	67.4	92.7
65 . . . 210 bpm	72.8	93.7
70 . . . 210 bpm	**73.6**	95.0
75 . . . 210 bpm	73.4	95.0
80 . . . 210 bpm	73.5	**95.1**
70 . . . 215 bpm	75.5	94.7
70 . . . 220 bpm	75.5	94.7
70 . . . 225 bpm	**76.4**	**95.1**
70 . . . 230 bpm	76.4	95.1

Table 5.11: Tempo induction performance of different tempo ranges for the autocorrelation function with a threshold $\theta_t = 0.075$. As inputs for the BLSTM network with three hidden layers with 25 units each, Mel spectrograms and positive median first order differences with window sizes of 23.22, 46.44, and 92.88 ms are used. Given are the Accuracies 1 and 2 for the BRD set.

The new tempo induction method was not only tested against ballroom dance music as present in the training set, but also with completely different music styles as present in the MTV data set. Since the tempo distribution of the MTV set (figure 4.1(b)) is different from the set used for training (most excerpts were taken from the BRD set), it was investigated how the results change if these tempo bounds are adapted to the range present in the MTV set. The upper limit is reduced down to 180 bpm and the lower limit down to 50 bpm. Whatever is changed with regard to the original tempo range, accuracy 2 decreases. Lowering the upper limit to the actual maximum tempo of the MTV set raised the accuracy 1 considerably. Adjustment of the lower bound down lowered the accuracy 1. From the numbers in table 5.12, it can be concluded that the best results can be obtained when using the tempo range of the training set, and constrain it whenever possible. Extending the range leads to degraded performance.

MTV [%]	Accuracy 1	Accuracy 2
70 . . . 225 bpm	73.9	**99.0**
70 . . . 200 bpm	79.4	97.5
70 . . . 180 bpm	**82.9**	97.0
65 . . . 180 bpm	78.9	94.0
60 . . . 180 bpm	76.9	92.0
55 . . . 180 bpm	71.4	90.7
50 . . . 180 bpm	68.3	90.5

Table 5.12: Tempo induction performance of different tempo ranges for the autocorrelation function with a threshold $\theta_t = 0.075$. As inputs for the BLSTM network with three hidden layers with 25 units each, Mel spectrograms and positive median first order differences with window sizes of 23.22, 46.44, and 92.88 ms are used. Given are the Accuracies 1 and 2 for the MTV set.

5.4 Remarks

It must be noted that all steps of the evaluation process highly interact with each other and depend on a lot of different parameters. It is therefore likely that changing one parameter could improve the given results further. Since experimenting with neural nets is often based on the trial and error method, one might take a semi optimal route at a certain point without recognising. To fully analyse the effect of all variables, a lot more evaluations must be performed. Since some of the training runs for neural networks take days, it was impossible to run more evaluation tasks in the given time frame.

Chapter 6

Results

In this chapter the best results obtained during the evaluation and testing process are compared to published results. Results are given separately for onsets, beats, and tempo detection. All used parameters are determined on the basis of the validation sets, the final results are given for the test sets and are obtained with exactly these parameters. This is essential to get results completely independent from the training and evaluation set.

Since the weights of the neural networks are initialised randomly, always ten complete 8-fold cross validation runs were performed. For determining the performance of the data sets, the mean of all output activations is calculated before classification.

6.1 Onset detection

Tables 6.1 to 6.4 show the results for the JPB set (see section 4.3). In [5] and [21], an onset is reported as correct, if it is recognised within a 100 ms window (±50 ms) around the annotated ground truth onset position. Since a resolution of 100 fps is used throughout all test, an onset could only be identified on a frame basis (10 ms length), which corresponds to an average error of ±5 ms and a maximum error of 10 ms, depending on whether the onset occurs on the beginning or end of the frame. Since this error is always present, the detection window set to 9 frames, which leads to a maximum overall error of 50 ms. This detection window is abbreviated ω_{100} in the ongoing.

In [13], a smaller window of ±25 ms is used for non-pitched percussive sounds. To show the precision of the neural network based approach, each table lists a second result for a smaller window of 5 frames, denoted with ω_{50}.

Each table in the following subsections list the results for the four categories of audio examples, and shows the performance for different onset detection functions: high frequency content (HFC), spectral difference (SD), spectral flux (SF), phase deviation

(PD), weighted phase deviation (WPD), normalised weighted phase deviation (NWPD), complex domain (CD), rectified complex domain (RCD), wavelet regularity modulus (WRM), negative log.-likelihood (NLL), and the new bidirectional Long-Short Term Memory recurrent neural network method (BLSTM). All conventional detection functions are briefly described in section 1.1, the new approach in section 3.1.

The best F-measure results for the traditional algorithms are highlighted in bold, as well as the ones for the new approach if they are better or on the same level.

6.1.1 PNP set

The PNP set has 93 onsets, but consists of only one audio file of string sounds. As a consequence, the results in table 6.1 are not that significant as the others. They can vary a lot, depending on the parameters used [21] or the underlaying sound material [13]. Nonetheless, the new methods shows results on par with the best performing algorithm (negative log.-likelihood), and is 1.5% better than the best spectrogram based approach (spectral flux). Since it is really difficult to annotate soft onsets with a very high accuracy, the result for the smaller detection window have to be seen with the awareness of this fact.

6.1.2 PP set

The published results [5, 21] for the PP set are based on the original test set with 489 onsets. The set has been modified by its author since then and has 482 onsets now. Thus, the BLSTM result in table 6.2 for the original annotations can be worse up to 1.4%. But even without this possible performance penalty, the new method is on par with the best performing traditional algorithm (spectral flux). This is not surprising at all, since it incorporates very similar signal features in its input signal representation as the spectral flux method.

6.1.3 NPP set

Non-pitched percussive sounds are the domain of the high frequency content onset detection method, since they contain a lot of noise in the higher frequencies. But this information is also present for the neural network and it can be concluded that it adapts quite well on this kind of sounds, too. Remarkable are the good results in table 6.3 for the smaller detection window ω_{50}, which demonstrate the high achievable precision of the new approach.

PNP	Precision	Recall	F-measure	TP[%]	FP[%]
HFC [5]	0.844	0.817	0.830	81.7	14.7
SD [5]	0.910	0.871	0.890	87.1	8.6
PD [5]	0.957	0.957	0.957	95.7	4.3
WRM [5]	0.905	0.925	0.915	92.5	10.1
NLL[5]	0.968	0.968	**0.968**	96.8	3.2
SF [21]	0.938	0.968	0.952	96.8	6.5
PD [21]	0.654	0.935	0.770	93.5	49.5
WPD [21]	0.937	0.957	0.947	95.7	6.5
NWPD [21]	0.909	0.968	0.938	96.8	9.7
CD [21]	0.946	0.946	0.946	94.6	5.4
RCD [21]	0.948	0.978	0.963	97.8	5.4
BLSTM $(orig, \omega_{100})$	0.968	0.968	**0.968**	96.8	3.2
BLSTM $(comb, \omega_{100})$	0.968	0.968	**0.968**	96.8	3.2
BLSTM $(comb, \omega_{50})$	0.939	0.939	0.939	93.9	6.1

Table 6.1: Onset detection results for the PNP set. The used onset annotations are marked *orig* and *comb*, the detection windows ω_{100} and ω_{50}. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the PNP test set of a 8-fold cross validation run on the onset set. Results taken from other publications are marked accordingly.

PP	Precision	Recall	F-measure	TP[%]	FP[%]
HFC [5]	0.947	0.941	0.944	94.1	5.4
SD [5]	0.983	0.949	0.966	94.9	1.6
PD [5]	0.996	0.955	0.975	95.5	0.3
WRM [5]	0.948	0.927	0.937	92.7	5.1
NLL[5]	0.968	0.924	0.945	92.4	3.1
SF [21]	0.981	0.988	**0.984**	98.8	1.8
PD [21]	0.482	0.865	0.619	86.5	93.0
WPD [21]	0.899	0.925	0.912	92.5	5.4
NWPD [21]	0.961	0.981	0.971	98.1	10.4
CD [21]	0.971	0.984	0.978	98.4	2.9
RCD [21]	0.983	0.979	0.981	97.9	1.6
BLSTM ($orig, \omega_{100}$)	0.987	0.987	**0.987**	98.7	1.3
BLSTM (mod, ω_{100})	0.992	0.992	**0.992**	99.2	0.8
BLSTM (mod, ω_{50})	0.983	0.979	0.981	97.9	1.7
BLSTM ($comb, \omega_{100}$)	0.986	0.993	**0.989**	99.3	1.4
BLSTM ($comb, \omega_{50}$)	0.974	0.986	0.980	98.6	2.6

Table 6.2: Onset detection results for the PP set. The used onset annotations are marked *orig*, *mod*, and *comb*, the detection windows ω_{100} and ω_{50}. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the PP test set of a 8-fold cross validation run on the onset set. Results taken from other publications are marked accordingly.

NPP	Precision	Recall	F-measure	TP[%]	FP[%]
HFC [5]	1.000	0.967	**0.983**	96.7	0.0
SD [5]	0.935	0.816	0.871	81.6	5.5
PD [5]	0.934	0.807	0.866	80.7	5.5
WRM [5]	0.974	0.887	0.928	88.7	2.2
NLL[5]	0.980	0.929	0.954	92.9	1.7
SF [21]	0.959	0.975	0.967	97.5	4.2
PD [21]	0.750	0.933	0.831	93.3	31.1
WPD [21]	0.974	0.958	0.966	95.8	2.4
NWPD [21]	0.950	0.966	0.958	96.6	5.2
CD [21]	0.948	0.924	0.936	92.4	5.2
RCD [21]	0.944	0.983	0.963	98.3	5.7
BLSTM ($orig, \omega_{100}$)	0.991	0.995	**0.993**	99.5	0.9
BLSTM (mod, ω_{100})	0.991	0.995	**0.993**	99.5	0.9
BLSTM (mod, ω_{50})	0.986	0.986	**0.986**	98.6	1.4
BLSTM ($comb, \omega_{100}$)	0.991	0.995	**0.993**	99.5	0.9
BLSTM ($comb, \omega_{50}$)	0.986	0.986	**0.986**	98.6	1.4

Table 6.3: Onset detection results for the NPP set. The used onset annotations are marked *orig*, *mod*, and *comb*, the detection windows ω_{100} and ω_{50}. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the NPP test set of a 8-fold cross validation run on the onset set. Results taken from other publications are marked accordingly.

6.1.4 MIX set

Although the results for the other sets were good as well, those for the MIX set in table 6.4 demonstrate the exceptional performance of the new approach. The gain is 3.6% absolute and 4.1% relative. Here the real potential of the new approach becomes apparent. Since details for the other detection methods are not available, it could only suspected that those methods mostly miss the quiet and thus hard to detect onsets. Since the new approach incorporates different representations of the input signal, it can therefore reveal details which might be hidden if only one spectrogram or the first order difference for a single STFT window length is used.

MIX	Precision	Recall	F-measure	TP[%]	FP[%]
HFC [5]	0.888	0.845	0.866	84.5	10.8
SD [5]	0.886	0.804	0.843	80.4	10.4
PD [5]	0.764	0.801	0.782	80.1	24.7
WRM [5]	0.768	0.819	0.793	81.9	24.7
NLL[5]	0.889	0.860	0.874	86.0	10.8
SF [21]	0.882	0.882	**0.882**	88.2	11.8
PD [21]	0.663	0.749	0.704	74.9	38.0
WPD [21]	0.843	0.830	0.836	83.0	15.5
NWPD [21]	0.916	0.845	0.879	84.5	7.7
CD [21]	0.941	0.819	0.876	81.9	5.2
RCD [21]	0.945	0.819	0.877	81.9	4.8
BLSTM $(orig, \omega_{100})$	0.941	0.897	**0.918**	89.7	5.6
BLSTM (mod, ω_{100})	0.947	0.930	**0.938**	93.0	5.3
BLSTM (mod, ω_{50})	0.921	0.896	**0.909**	89.6	7.9
BLSTM $(comb, \omega_{100})$	0.938	0.918	**0.928**	91.8	6.2
BLSTM $(comb, \omega_{50})$	0.907	0.878	**0.893**	87.8	9.3

Table 6.4: Onset detection results for the MIX set. The used onset annotations are marked *orig*, *mod*, and *comb*, the detection windows ω_{100} and ω_{50}. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the MIX test set of a 8-fold cross validation run on the onset set. Results taken from other publications are marked accordingly.

6.1.5 Onset set

All the above mentioned sets are relatively small, so the results for the complete onset set are given in table 6.5. Since the vast amount of this set is complex music, the results are very similar to the ones obtained with the MIX set. The complete annotations contain 6605 onsets, if all onsets within 30 ms are combined, the number of onsets is reduced to 5861. Again, the results are on a very high level, far ahead of the best reported results so far.

Onset	Precision	Recall	F-measure	TP[%]	FP[%]
BLSTM $(mod, \omega_{100}, \lambda_g)$	0.945	0.925	0.935	92.5	5.5
BLSTM $(mod, \omega_{50}, \lambda_g)$	0.920	0.901	0.911	90.1	8.0
BLSTM $(comb, \omega_{100}, \lambda_g)$	0.938	0.916	0.927	91.6	6.2
BLSTM $(comb, \omega_{50}, \lambda_g)$	0.911	0.890	0.900	89.0	8.9

Table 6.5: Onset detection results on the complete onsets test set. The modified onsets are marked *mod*, and the combined onset *comb*. The used detection windows with sizes of 100 and 50 ms are abbreviated ω_{100} and ω_{50}. Given are the precision, recall, and F-measure as well as the percentage of true positives (TP) and false positives (FP) for the test set of a 8-fold cross validation run on the onset set.

6.1.6 Discussion

The results for the given sets show clearly that the new onset detection method is state-of-the art. It achieves the best performances reported so far, independently from the type of music or onset.

One of the main advantages of the new approach is the simplified thresholding process. This is due to the fact that the output of the neural network produces sharp peaks without a broad noise floor. This can be seen in figure 6.1, which compares the new BLSTM method to the Spectral Flux onset detection method.

The new method not only makes adaptive thresholds obsolete, it can be operated with global thresholding settings as well. All results in table 6.1 to 6.4 are obtained with a global threshold scaling factor $\lambda_o = 50$ instead of adjusted threshold parameters for each set, as used in [5] and [21].

If the threshold scaling factor λ was determined individually for each set (based on the corresponding part of the validation set), the results deviated only up to a maximum of 1% absolute, but most of the times less than 0.2%. Hence no additional results with individual threshold scaling factors are given.

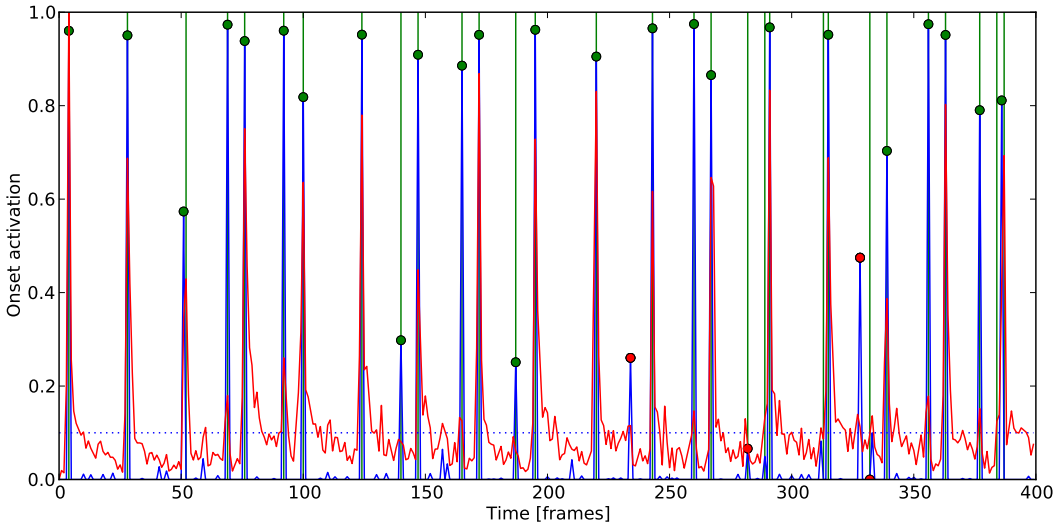


Figure 6.1: Onset functions of different detection methods: the Spectral Flux method (red), the BLSTM method (blue), ground truth onsets (green), correctly identified onsets (green dots), missed or false onsets (red dots), and a fixed threshold $\theta_o = 0.1$ (blue dotted) for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*.

There don't exist any comparable precision, recall, or F-measure values for the complete JPB set, but the ROC curve in figure 6.2 gives an estimation of how far the performance of the new approach is ahead of all other methods tested in [5], if global thresholding parameters are used.

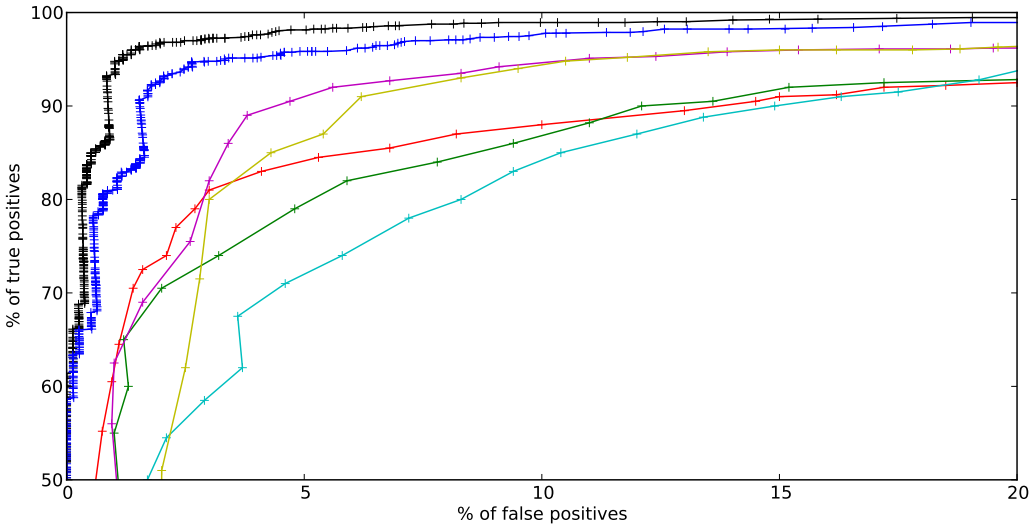


Figure 6.2: Performance comparison of onset detection algorithms on the JPB set. The new BLSTM approach with wide (black) and small detection window (blue). Alternative methods (values extracted from [5]): HFC (yellow), SD (red), PD (green), WRM (cyan), and NLL (yellow).

6.2 Beat detection

Table 6.6 shows the result for beat detection on the complete beat test set, consisting of 2057 annotated beats. A ω_{150} detection window of size 150 ms ($\pm$75 ms) is used to distinguish whether beats are detected correctly or not. Due to the lack of a publicly available test set, comparable results are missing completely.

Beats	Precision	Recall	F-measure	TP[%]	FP[%]
BLSTM	0.813	0.808	0.810	80.8	18.7

Table 6.6: Beat detection results on the complete beat test set. The detection window ω_{150} with a size of 150 ms is used. Given are the precision, recall, and F-measure rates as well as the percentage of true (TP) and false positives (FP).

Nonetheless some conclusions can be drawn from the results of the the tempo induction algorithm, since this is based completely on the detected beats. If other methods would detect the beats as reliable as the new proposed method, tempo induction results in the same range as those outlined in the next section could be expected. Since those do not exist, it is conjectured that the new beat detection method works better than existing ones on the used sets.

6.3 Tempo induction

Tempo induction performance is tested with two sets, the BRD and the MTV set. They contain very different types of music, namely Ballroom dance music with a wide spread tempo distribution, and Pop music with a more compact tempo distribution concentrated around the range of 100-120 bpm.

6.3.1 BRD set

Table 6.7 shows the performance results of different tempo induction functions for the BRD set. The first is the result of the winner of the tempo induction contest held during ISMIR 2004 [44], Klapuri. Additionally three different methods published by Schuller et. al. in [83] are given. One is without ballroom dance style recognition, another incorporates the detected ballroom dance style, and the last the given ground-truth ballroom dance style. As comparison two different results for the new neural network based approach are given. The first represents the tempo detection performance of the algorithm described in section 3.3, and the second shows the results if the ground truth

ballroom dance style is included to constrain the searched tempo range (like gt BDS of Schuller et. al.).

BRD	Accuracy 1[%]	Accuracy 2[%]
Klapuri [44]	63.2	**91.0**
Schuller et. al. (w/o BDS) [83]	**69.8**	88.8
Schuller et. al. (w BDS) [83]	86.9	93.0
Schuller et. al. (gt BDS) [83]	92.4	92.8
BLSTM	**75.2**	**95.1**
BLSTM (gt BDS)	92.8	94.4

Table 6.7: Results on BRD set. The used tempo range for the BLSTM method is 75 bpm to 220 bpm, with a threshold $\theta_t = 0.075$ for the ACF of the beat activation function. Given are the percentages of instances with correctly identified tempo without (Accuracy 1), and with octave errors included (Accuracy 2).

It can be seen that the results obtained with the tempo induction method based on the new neural network beat detector outperforms the existing methods by 5.4% in accuracy 1 and 4.1% in accuracy 2. This is a remarkable performance boost. Even in the case of the somewhat synthetic benchmark with ground truth ballroom dance style included it outperforms the other methods, although the gain is much smaller.

6.3.2 MTV set

Although not trained with modern pop music, the new tempo induction method was also tested with that kind of music present in the MTV set. Table 6.8 shows the achieved results and compares them the one published in [30]. Depending on whether the tempo range is adjusted, different conclusions can be drawn. First of all, the new approach demonstrates adequate performance (with an exceptional Accuracy 2 of 99%) with completely unknown music, if the same tempo range as for training is used. If the tempo range is constrained (but not extended) to the actual tempo present in the tested set, previously unachieved Accuracy 1 values can be achieved. The gain is 8.9% absolute and 12% relative.

6.3.3 Discussion

The results for both the BRD and MTV set show that the new approach boosts the tempo induction performance considerably. Since the calculation of the tempo is solely

MTV	Accuracy 1[%]	Accuracy 2[%]
Eyben [30]	74.0	97.0
BLSTM (70 . . . 225 bpm)	73.9	**99.0**
BLSTM (70 . . . 180 bpm)	**82.9**	97.0

Table 6.8: Tempo detection results on MTV set with different tempo ranges. For the ACF of the beat activation function, a threshold $\theta_t = 0.075$ was used. Given are the percentages of instances with correctly identified tempo without (Accuracy 1), and with octave errors included (Accuracy 2).

based on the detected beats, the same can be concluded for the beat detection. One of the most obvious reasons can be seen in figure 6.3: the BLSTM approach can reliably detect the beats (coinciding with the quarter notes), whereas spectral feature based autocorrelation and comb filter methods often detect the eighth notes as well. The ability to distinguish between regions with high spectral energy (refer to figures 5.2 and 5.5 for the spectrograms and positive differences of the underlaying audio signal) is definitely one of the biggest advantages of the neural network based approach.

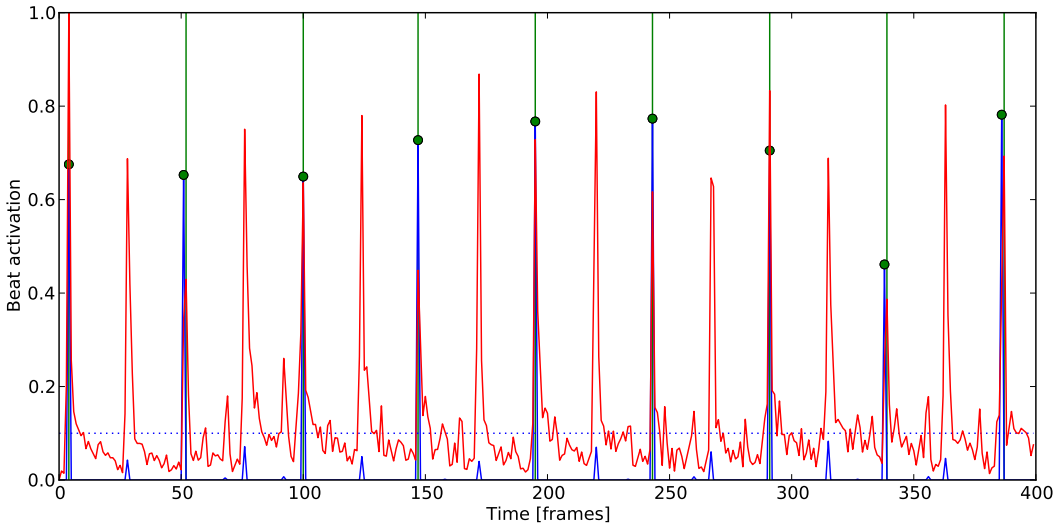


Figure 6.3: Beat functions of different detection methods: spectral based method (red), the BLSTM method (blue), ground truth beats (green), correctly identified beats (green dots), and a fixed threshold $\theta_b = 0.1$ (blue dotted) for a 4 seconds excerpt from *Basement Jaxx - Rendez-Vu*.

Chapter 7

Conclusion and Outlook

7.1 Conclusion

This work introduced a new approach for onset and beat detection as well as tempo induction. It shows performance on par with or better than all existing state-of-the-art approaches. It is completely data driven and works solely on the given input signal, without any higher level knowledge.

With the use of the neural network stage instead of traditional reduction functions, the onset detector is capable of combining different techniques present in these reduction functions to a perfectly working system. It achieves fantastic performance results independently from the signal type, thus making the choice of a suitable onset detection for a given sound obsolete. Furthermore it is capable of detecting onsets with an extremely high temporal precision, which makes the new onset detector perfectly suitable for music transcription tasks. An overall performance gain of 3.0% F-measure absolute (4.1% relative) is observed for complex music mixes. The average improvement on the whole JPB data set is 1.7% F-measure absolute. It is not necessary to change any parameters to achieve these good results. This is an extremely important step towards a universally deployable onset detector.

Although trained with a much lesser amount of annotated beats, the beat detector shows exceptional results as well. Traditional beat detectors often work as a second stage after the tempo induction, trying to estimate the phase of the detected tempo. They are therefore limited to a more or less constant beat. Another limitation of these algorithms is the often made assumption that the beats occur at strict metrical levels, mostly a multiplicative of the smallest measure called the tatum. The new neural network based approach lifts these limitations. Once again it operates solely on the signal without any higher level knowledge about beat or rhythm structures. It is therefore capable of tracking beats even if the tempo changes rapidly.

The proposed tempo induction method uses the detected beats to calculate the tempo

on basis of the most dominant inter beat interval. Reliably detected beats are the basis of a much better tempo estimation compared to existing methods. Especially the performance of correctly identified tempo raises about 5.4% and 8.9% absolute (7.7% and 12% relative) depending on the used data set. If not the whole piece of music, but only a fraction is considered, tempo changes can be detectable as well. Since the used music excepts were of constant tempo, this could not be tested however.

Limitations

Despite all the positive aspects of the new proposed algorithm, it has a few downsides, too. First of all, the method is not able to operate online, as it needs all data to be present before processing starts, due to the use of bidirectional LSTM recurrent neural networks. If online operation is necessary, a simple LSTM recurrent network can be used as well. In case of the onset detector, the performance penalty is less than 1% on average, but for the beat detector the penalty is more than 10%.

Another aspect to be mentioned is that the new approach has a very high computational complexity. Taking not only one, but two or even three STFT spectrograms and their first order differences adds an overhead compared to traditional spectral feature based algorithms. Also the neural network stage adds some extra amount of processing time. Nonetheless, all computation can be done in real-time on commodity hardware[1].

Last but not least, precisely annotated ground truth data is needed for the training of the neural networks. The more annotated material exists, the better. This can be seen at the differences in beat detection for one and the same file shown in figure 5.6, depending on how the set is split into training and validation set. Such variations could not be observed during onset detection runs. Taking apart the possibility that beats are more complex to be learned by the neural network, it could be concluded that more training material is needed to achieve better and more reliable results.

7.2 Outlook

Beside the general solution of greater annotated data sets, a few other areas for possible improvements and further development come to mind.

If the onsets are labelled more specifically (i.e. the types of instruments, vocals, etc.), and this information is combined with other (probably also neural network based [94])

[1] The author used a 2 GHz Intel Core 2 Duo with 4GB of memory.

methods for key and pitch detection, a complete music transcription system can be constructed.

With the information of the detected onsets and beats, more sophisticated classifiers for all kinds of temporal higher level knowledge and characteristics can be built. This includes, but is not limited to, dance style, meter, rhythm patterns, genre, etc..

Other input representations

In [26], a better onset detection precision is discovered, if a multi-resolution analysis is used instead of fixed one. It could be tested, whether the constant Q transform [10] can perform as good as or better than the actual proposed solution with combinations of different spectrograms. The constant Q transform has been applied successfully to other tasks, such as pitch and note detection [14], too.

Another possible enhancement could be achieved if a time-frequency representation of the signal is used instead of the spectrogram of the short time Fourier transform. The STFT contains only a perfect spectral resolution but no time information at all. Wavelet transforms [50] include both kind of informations and are already used in other methods for onset and beat detection [16, 100, 99].

Since the performed input normalisation (to mean 0 and standard deviation 1) of the input signal for the neural network had a negative effect on the overall performance, it was not considered in this thesis. It could be analysed, whether a normalisation based on Moore's model of loudness [76], as used by [59] and [97] gives an improvement.

Neural network modifications

As mentioned earlier, it is favourable to have big training sets. But using the early stopping method always requires to sacrifice part of the annotated set as validation set. Training with noise [9] is a method to circumvent this. However, since it does not lead to better performance in all cases and the amount of added noise or jitter largely depends on the used data, the time consuming experiments to determine the best amount relative to the input data have not been performed.

For onset or beat detection, a special classification algorithm for the output activation function of the neural network is used. This does not only include a different peak detection function instead of the standard soft-max, but also an error function used to determine whether an onset or beat is identified correctly. During training the neural network also uses the amount of error to adjust the weights of the connections, but it uses the very simple cross entropy error. Changing this error calculation function to a

more sophisticated one similar to the one used for identifying the onsets or beats might improve the results further. For example, if an onset recognised one frame apart from the annotated position would not count as a false detected one, but would tell the neural network that the overall tendency is good and it misses just the last amount of precision, could help to enhance the potential of the neural network.

Finally, it could be investigated, whether the performance of the LSTM neural net without forget gates and peep holes (the connections from the memory cell to the gates in figure 2.3) improves the results for music information retrieval as it does in other areas like signature verification tasks [98].

Closing words

In this thesis a new state-of-the-art onset, beat, and tempo detection method was proposed. It is purely data driven and works without incorporating any higher level knowledge about the underlaying audio material. Furthermore, it shows exceptional results without the need of adjusting any parameters to the type of music.

Bibliography

[1] P. E. Allen and R. E. Dannenberg. Tracking musical beats in real time. In *International Computer Music Conference*, pages 140–143. International Computer Music Association, September 1990.

[2] M. Alonso, G. Richard, and B. David. Tempo and beat estimation of musical signals. In *ISMIR*, 2004.

[3] M. Alonso, G. Richard, and B. David. Accurate tempo estimation based on harmonic + noise decomposition. *EURASIP Journal on Applied Signal Processing*, pages 161–161, 2007.

[4] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall, Inc., 1993.

[5] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, September 2005.

[6] J. Bello, C. Duxbury, M. Davies, and M. Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, June 2004.

[7] J. Bello, E. Ravelli, and M. Sandler. Drum sound analysis for the manipulation of rhythm in drum loops. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume 5, pages 233–236, May 2006.

[8] J. Bello and M. Sandler. Phase-based note onset detection for music signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*, volume 5, pages 441–444, April 2003.

[9] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[10] J. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant q transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701, November 1992.

[11] I. Bruno, S. Monni, and P. Nesi. Automatic music transcription supporting different instruments. In *Proceedings of the International Conference on Web Delivering of Music (WEDELMUSIC 2003)*, pages 37–44, September 2003.

[12] C.-H. Chuan and E. Chew. The effect of key and tempo on audio onset detection using machine learning techniques: A sensitivity analysis. In *Proceedings of the IEEE International Symposium on Multimedia (ISM 2006)*, pages 805–810, December 2006.

[13] N. Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *Proceedings of the AES Convention 118*, pages 28–31, 2005.

[14] G. Costantini, M. Todisco, and R. Perfetti. On the use of memory for detecting musical notes in polyphonic piano music. In *Proceedings of the European Conference on Circuit Theory and Design (ECCTD 2009)*, pages 806–809, August 2009.

[15] A. D'Aguanno and G. Vercellesi. Tempo induction algorithm in mp3 compressed domain. In *Proceedings of the International Workshop on Multimedia Information Retrieval (MIR '07)*, pages 153–158, 2007.

[16] L. Daudet. Transient modelling by pruned wavelet trees. In *Proceedings of the International Computer Music Conference (ICMC)*, Sept. 2001.

[17] M. Davies and M. Plumbley. Beat tracking with a two state model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2005)*, volume 3, pages 241–244, March 2005.

[18] M. Davy and S. Godsill. Detection of abrupt spectral changes using support vector machines - an application to audio signal segmentation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, pages 1313–1316, 2002.

[19] S. Dixon. Beat induction and rhythm recognition. In *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pages 311–320, 1997.

[20] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[21] S. Dixon. Onset detection revisited. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-06)*, pages 133–137, Montreal, Quebec, Canada, September 2006.

[22] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *ISMIR*, 2004.

[23] S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In *ISMIR*, 2003.

[24] J. S. Downie. Music information retrieval. In *Annual Review of Information Science and Technology*, number 37, chapter 7, pages 295–340. Information Today Books, 2003.

[25] C. Duxbury, J. P. Bello, M. Davies, M. Sandler, and M. S. Complex domain onset detection for musical signals. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-03)*, 2003.

[26] C. Duxbury, J. P. Bello, M. Sandler, M. S, and M. Davies. A comparison between fixed and multiresolution analysis for onset detection in musical signals. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-04)*, 2004.

[27] C. Duxbury, M. Sandler, and M. Davis. A hybrid approach to musical note onset detection. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-02)*, September 2002.

[28] D. Eck. Beat tracking using an autocorrelation phase matrix. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1313–1316, April 2007.

[29] D. Ellis and G. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1429–1432, April 2007.

[30] F. Eyben. High level rhythmic audio features for robust music information retrieval. Bachelor's thesis. Technische Universität München, 2006.

[31] F. Eyben, B. Schuller, S. Reiter, and G. Rigoll. Wearable assistance for the ballroom-dance hobbyist - holistic rhythm analysis and dance-style classification. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2007)*, pages 92–95, July 2007.

[32] H. Fastl and E. Zwicker. *Psychoacoustics: Facts and Models*. Springer-Verlag New York, Inc., 2006.

[33] J. Foote, M. L. Cooper, and U. Nam. Audio retrieval by rhythmic similarity. In *ISMIR*, 2002.

[34] J. Foote and S. Uchihashi. The beat spectrum: a new approach to rhythm analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2001)*, pages 881–884, August 2001.

[35] S. Gao and C.-H. Lee. An adaptive learning approach to music tempo and beat analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, volume 4, pages 237–240, May 2004.

[36] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[37] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proceedings of the ACM International Conference on Multimedia (MULTIMEDIA '94)*, pages 365–372, 1994.

[38] M. Goto and Y. Muraoka. Beat tracking based on multiple-agent architecture a real-time beat tracking system for audio signals. In *Proceedings of the International Conference on Multiagent Systems*, pages 103–110, 1996.

[39] M. Goto and Y. Muraoka. Real-time beat tracking for drumless audio signals: chord change detection for musical decisions. *Speech Communication*, 27:311–335, 1999.

[40] F. Gouyon, S. Dixon, E. Pampalk, and G. Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th international Conference*, June 2004.

[41] F. Gouyon, S. Dixon, and G. Widmer. Evaluating low-level features for beat classification and tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1309–1312, April 2007.

[42] F. Gouyon, L. Fabig, and J. Bonada. Rhythmic expressiveness transformations of audio recordings swing modifications. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-03)*, London, September 2003.

[43] F. Gouyon, P. Herrar, and P. Herrera. A beat induction method for musical audio signals. In *Proceedings of the 4th WIAMIS Special Session on Audio Segmentation and Digital Music*, 2003.

[44] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, September 2006.

[45] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technische Universität München, 2008.

[46] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.

In *Proceedings of the International Conference on Machine learning (ICML '06)*, pages 369–376, 2006.

[47] A. Graves, S. Fernández, M. Liwicki, H. Bunke, and J. Schmidhuber. Unconstrained on-line handwriting recognition with recurrent neural networks. In *NIPS*, 2007.

[48] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855–868, 2009.

[49] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *IEEE Transactions on Neural Networks*, 18:602–610, 2005.

[50] A. Haar. Zur Theorie der Orthogonalen Funktionensysteme. In *Mathematische Annalen*, volume 69, pages 331–371. Springer-Verlag, 1910.

[51] S. Hainsworth and M. Macleod. Beat tracking with particle filtering algorithms. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 91–94, October 2003.

[52] S. W. Hainsworth. *Techniques for the Automated Analysis of Musical Audio*. PhD thesis, University of Cambridge, UK, September 2004.

[53] S. Handel. *Listening: an introduction to the perception of auditory events*. MIT Press, 1989.

[54] R. Harper and M. Jernigan. Self-adjusting beat detection and prediction in music. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, volume 4, pages 245–248, May 2004.

[55] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computing*, 9(8):1735–1780, 1997.

[56] A. Holzapfel and Y. Stylianou. Rhythmic similarity of music based on dynamic periodicity warping. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 2217–2220, April 2008.

[57] Ismir 2004 ballroom data set. `http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html`, 2004.

[58] E. Kapanci and A. Pfeffer. A hierarchical approach to onset detection. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 438–441, 2006.

[59] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1999)*, volume 6, pages 3089–3092, March 1999.

[60] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, January 2006.

[61] A. Lacoste and D. Eck. Onset detection with artificial neural networks. MIREX note onset detection contest, 2005.

[62] A. Lacoste and D. Eck. A supervised classification algorithm for note onset detection. *EURASIP Journal on Applied Signal Processing*, (1):153–153, 2007.

[63] J. Laroche. Estimating tempo, swing and beat locations in audio recordings. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 135–138, 2001.

[64] J. Laroche. Efficient tempo and beat tracking in audio recordings. In *Journal of the Audio Engineering Society*, volume 51, pages 226–233. Audio Engineering Society, April 2003.

[65] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient BackProp. *Lecture Notes in Computer Science*, 1524, 1998.

[66] M. Li and T. Li. Pitch recognition based on intelligent neural network system. In *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS 2004)*, volume 2, pages 1081–1085, June 2004.

[67] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 143–146, October 2003.

[68] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification, 2005.

[69] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval*, 2000.

[70] M. Marolt. Transcription of polyphonic piano music with neural networks. In *Proceedings of the Mediterranean Electrotechnical Conference (MELECON 2000)*, volume 2, pages 512–515, 2000.

[71] M. Marolt, A. Kavcic, M. Privosnik, and S. Divjak. On detecting note onsets in piano music. In *Proceedings of the Mediterranean Electrotechnical Conference (MELECON 2002)*, pages 385–389, 2002.

[72] P. Masri. *Computer Modeling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, University of Bristol, UK, 1996.

[73] Mirex 2009 beat tracking contest. `http://www.music-ir.org/mirex/2009/index.php/Audio_Beat_Tracking_Results`, 2009.

[74] K. Miyamoto, H. Kameoka, H. Takeda, T. Nishimoto, and S. Sagayama. Probabilistic approach to automatic music transcription from audio signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 2, pages 697–700, April 2007.

[75] H. Miyao and Y. Nakano. Head and stem extraction from printed music scores using a neural network approach. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 2, pages 1074–1079, August 1995.

[76] B. C. J. Moore, B. R. Glasberg, and T. Baer. A model for the prediction of thresholds, loudness, and partial loudness. In *Journal of the Audio Engineering Society*, volume 45, pages 224–240. Audio Engineering Society, April 1997.

[77] J. Paulus and A. Klapuri. Measuring the similarity of rhythmic patterns. In *ISMIR*, 2002.

[78] G. Peeters. Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume 5, pages 53–56, May 2006.

[79] D. Ruck, S. Rogers, M. Kabrisky, M. Oxley, and B. Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, December 1990.

[80] N. Scaringella and G. Zoia. A real-time beat tracker for unrestricted audio signals. In *Proceedings of the Actes des Journees d'Informatique Musicale (JIM2004)*, 2004.

[81] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[82] B. Schuller, F. Eyben, and G. Rigoll. Fast and robust meter and tempo recognition for the automatic discrimination of ballroom dance styles. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 1, pages 217–220, April 2007.

[83] B. Schuller, F. Eyben, and G. Rigoll. Tango or waltz?: putting ballroom dance style into tempo detection. *EURASIP Journal of Audio Speech Music Processing*, pages 1–12, 2008.

[84] J. Seppänen. Computational models of musical meter recognition. Master's thesis, Tampere University of Technology, 2001.

[85] W. Sethares and R. Arora. Equilibria of adaptive wavetable oscillators with applications to beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1301–1304, April 2007.

[86] W. Sethares, R. Morris, and J. Sethares. Beat tracking of musical performances using low-level audio features. *Speech and Audio Processing, IEEE Transactions on*, 13(2):275–285, March 2005.

[87] Y. Shiu, N. Cho, P.-C. Chang, and C.-C. Kuo. Robust on-line beat tracking with kalman filtering and probabilistic data association (kf-pda). *IEEE Transactions on Consumer Electronics*, 54(3):1369–1377, August 2008.

[88] Y. Shiu and C.-C. Kuo. A modified kalman filtering approach to on-line musical beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 2, pages 765–768, April 2007.

[89] Y. Shiu and C.-C. Kuo. On-line musical beat tracking with phase-locked-loop (pll) techniique. In *Proceedings of the International Conference on Consumer Electronics (ICCE 2007)*, pages 1–2, January 2007.

[90] P. Smaragdis. Non-negative matrix factor deconvolution; extracation of multiple sound sources from monophonic inputs. In *Independent Component Analysis and Blind Signal Separation*, volume 3195 of *Lecture Notes in Computer Science (LNCS)*, pages 494–499. Springer Verlag, October 2004.

[91] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, 2003.

[92] L. Smith and D. Fraser. Robust sound onset detection using leaky integrate-and-fire neurons with depressing synapses. *IEEE Transactions on Neural Networks*, 15(5):1125–1134, September 2004.

[93] D. Stowell and M. Plumbley. Adaptive whitening for improved real-time audio onset detection, 2007.

[94] J. Sun, H. Li, and L. Lei. Key detection through pitch class distribution model and ann. In *Proceedings of the IEEE International Conference on Digital Signal Processing (DSP 2009)*, pages 1–6, July 2009.

[95] J. Sundberg. *The Science of Musical Sounds*. Academic Press, San Diego, 1991.

[96] H. Takeda, T. Nishimoto, and S. Sagayama. Rhythm and tempo analysis toward automatic music transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1317–1320, April 2007.

[97] B. Thoshkahna and K. Ramakrishnan. A psychoacoustics based sound onset detection algorithm for polyphonic audio. In *Proceedings of the International Conference on Signal Processing (ICSP 2008)*, pages 1424–1427, October 2008.

[98] C. Tiflin and C. W. Omlin. LSTM recurrent neural networks for signature verification. In *Proceedings of the Southern African Telecommunication Networks and Applications Conference (SATNAC)*, 2003.

[99] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[100] G. Tzanetakis, G. Essl, and P. Cook. Audio analysis using the discrete wavelet transform. In *Proceedings of the IEEE Conference in Acoustics and Music Theory Applications*, 2001.

[101] Y. Wang and M. Vilermo. A compressed domain beat detector using mp3 audio bitstreams. In *Proceedings of the ACM International Conference on Multimedia (MULTIMEDIA '01)*, pages 194–202, 2001.

[102] V. Zenz and A. Rauber. Automatic chord detection incorporating beat and key detection. In *Proceedings of the IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*, pages 1175–1178, November 2007.

List of Figures

List of Tables

List of Symbols

List of Abbreviations

Appendix A

Mel filter bank calculation

If a STFT spectrogram needs to be transformed from the linear Hz frequency scale to the logarithmic spaced Mel scale, a filter bank needs to be constructed. The spectrogram is then multiplied with this filter bank to obtain the Mel spectrogram.

Conversion between the Hz frequency scale on the Mel frequency scale is done according to equations A.1 and A.2:

$$f_{Mel} = 1127 \cdot \ln\left(1 + \frac{f_{Hz}}{700}\right) \tag{A.1}$$

$$f_{Hz} = 700 \cdot \left(e^{f_{Mel}/1127} - 1\right) \tag{A.2}$$

Depending on the upper and lower frequencies f_{min} and f_{max} in Mel and the number of Mel bands M, which are spread equidistant over the Mel frequency scale, the frequency width f_{width} of each filter is determined as:

$$f_{width} = \frac{f_{max} - f_{min}}{M + 1} \tag{A.3}$$

and the centre frequencies $f_c(m)$ for each filter, with m being the number of the Mel frequency bin as:

$$f_c(m) = f_{min} + f_{width} \cdot m \tag{A.4}$$

With the help sampling frequency f_s and the number of STFT frequency bins K, these frequencies are converted back to the corresponding centre frequency bin numbers $k_c(m)$ according to this equation:

$$k_c(m) = f_c(m) \cdot \frac{K}{f_s} \tag{A.5}$$

The complete Mel filter bank is then constructed according the following rule:

$$\widetilde{F}(m,k) = \begin{cases} \frac{k-k_c(m-1)}{k_c(m)-k_c(m-1)} & \text{for } k_c(m-1) < k \leq k_c(m) \\ 1 - \frac{k-k_c(m)}{k_c(m+1)-k_c(m)} & \text{for } k_c(m) < k < k_c(m+1) \\ 0 & \text{otherwise} \end{cases} \tag{A.6}$$

Since the filters get wider with increasing Mel frequency bin numbers, (see figure A.1) each triangular filter is scaled by a factor κ, so that the energy of all filters is equal, i.e. the following condition is met:

$$\forall m \sum_{k=1}^{k=K} \widetilde{F}(m,k) = 1 \tag{A.7}$$

Each numerator of equation A.6 is therefore multiplied by the factor $\kappa = \frac{2}{k_c(m+1)-k_c(m-1)}$, and the resulting normalised filter bank is given by:

$$F(m,k) = \begin{cases} \frac{2(k-k_c(m-1))}{(k_c(m+1)-k_c(m-1))\cdot(k_c(m)-k_c(m-1))} & \text{for } k_c(m-1) < k \leq k_c(m) \\ \frac{2(k_c(m+1)-k)}{(k_c(m+1)-k_c(m-1))\cdot(k_c(m+1)-k_c(m))} & \text{for } k_c(m) < k < k_c(m+1) \\ 0 & \text{otherwise} \end{cases} \tag{A.8}$$

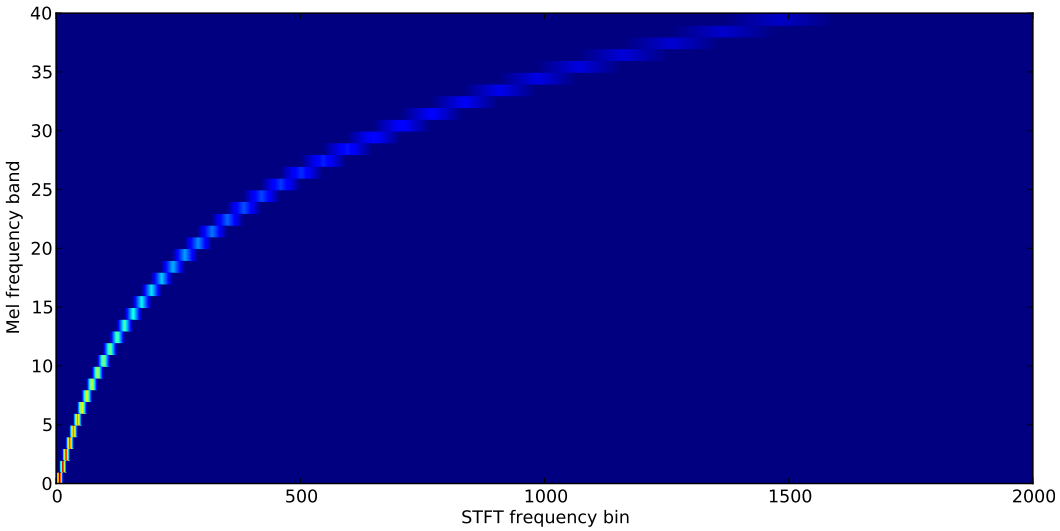


Figure A.1: Mel filter bank with 40 Mel bands and 2000 STFT frequency bins, for a frequency range of 20 Hz to 16.0 kHz.

Appendix B

Software

All software written for this thesis can be found on the software DVD or is available at the authors page located at `http://mir.minimoog.org`.

RNNLIB

Alex Graves wrote a neural network software called RNNLIB for his PhD thesis [45]. It was publicly available at github[1], but the author removed it later, but grants access to if asked. This software is used for all neural network tasks, namely training and testing.

Python

All other programming tasks, such as signal processing, transforming the input to the NetCDF format[2] needed by RNNLIB is performed with the Python programming language[3] and the numerical python extension[4].

Sonic Visualiser

For the onset and beat annotation task, the Sonic Visualiser[5] software developed at the Queen Mary University of London is used. The big advantage of that software is that several spectrograms can be viewed simultaneously, and the playback speed can be adjusted in a wide range, thus simplifying the task of annotating. All provided annotations are given in pure text format versions with one entry per line as well as `.sv` files, containing all annotations for onsets and beats.

[1] `http://wiki.github.com/alexgraves/RNNLIB`
[2] `http://www.unidata.ucar.edu/software/netcdf`
[3] `http://www.python.org`
[4] `http://numpy.scipy.org`
[5] `http://www.sonicvisualiser.org`