Singing Voice Extraction from 2-Channel Polyphonic Musical Recordings

Diploma Thesis

Sebastian Rieck

Supervisor: Dipl.-Ing. Dr.techn. Alois Sontacchi
Assessor: O.Univ.Prof. Mag.art. DI Dr.techn. Robert Höldrich

Graz, April 2012

Contents

Acknowledgements

I would like to take this opportunity to say „Thank You“ to the people who supported and accompanied me along the way.

First and foremost, to my family, for their continous support and love! To my sisters Elisa and Céline, for always having faith in me, i will always watch your back! To my parents, for knowing i can always count on them, for supporting me no matter what and for giving me the opportunity to chose my way.

To my supervisor, Alois Sontacchi, for giving me inspiration and guidance.

To the whole IEM Staff, for creating and sustaining a friendly and inspiring learning enviroment, where a helping hand is never out of reach.

And finally, to all my friends and fellow students, for every moment spent, from fun to inspiring conversations! I am shure we will meet again!

Abstract

This thesis deals with the extraction of singing voice signals from 2-channel polyphonic musical recordings. The proposed method consists of 2 steps. First, the assumption is used that singing voice is very likely positioned in the center of the stereo panorama. Using a similarity measure, this „center“ part is extracted and the resulting monophonic signal represents the basis for the subsequent processing. Second, voiced singing voice and unvoiced singing voice are extracted separately and summed up in a final step, to form the extracted singing voice. The extraction of the voiced singing voice is realized by detecting the fundamental frequency f_0 of singing voice, along with its corresponding partials. Using a sinusoidal model, all partial frequencies are then synthesized. The extraction of the unvoiced singing voice is realized by segmenting the monophonic signal in Time-Frequency-Units. Those that can be associated with singing voice are extracted.

Preprocessing the stereophonic recording, improves the accuracy of the voiced singing voice extraction process by 12%, and the accuracy in extracting the unvoiced singing by 7%. The detection of the singing voice f_0 is based on the Diploma Thesis of A. Rahimzadeh. We propose modifications and results show, that the average accuracy in detecting the f_0 is improved by 16%. The proposed method to extract singing voice has been evaluated using blind source separation performance measures and yields a average Source to Distortion Ratio of 35.1dB, which is an improvement of 5-10dB compared to state of the art methods. The average Source to Distortion Ratio results in -2.4dB.

Kurzfassung

Ziel dieser Arbeit ist die Extraktion der Gesangsstimme aus einer polyphonen Stereoaufnahme. Die vorgeschlagene Umsetzung besteht aus 2 Verarbeitungsschritten. Zunächst wird die Annahme verwendet, dass sich die Gesangsstimme in der Mitte des Stereopanoramas befindet. Unter Verwendung eines Ähnlichkeitsmaßes wird jener Teil aus der Stereoaufnahme extrahiert. Das resultierende Mono-Signal stellt das Ausgangssignal für die weitere Verarbeitung dar. Zweitens, wird der stimmhafte und der stimmlose Anteil des Gesangs separat extrahiert. Der extrahierte Gesang setzt sich dann aus der Summation beider genannten Anteile zusammen. Die Extraktion des stimmhaften Anteils beruht auf der Detektion der Grundfrequenz f_0 des Gesangs und der dazugehörigen Obertöne. Unter Verwendung des „Sinusoidal Model“ werden alle Partialtöne des Gesangs synthetisiert. Zur Extraktion des stimmlosen Anteils des Gesangs, wird das Mono-Signal in der Zeit-/Frequenzdomäne in Segmente unterteilt. Jene Segmente welche dem Gesang zugeordnet werden können, werden dann extrahiert.

Die Vorverarbeitung des Stereo-Signals verbessert die mittlere Genauigkeit der Extraktion des stimmhaften Gesangs um 12% und jene des stimmlosen Gesangs um 7%. Die Detektion der Gesangs f_0 basiert auf der Diplomarbeit von A. Rahimzadeh. Die vorgeschlagenen Änderung verbessern die mittlere Genauigkeit um 16%. Die Qualität der Extraktion der Gesangsstimme wurde mit Hilfe üblicher Maße evaluiert und erreicht eine mittlere „Source to Distortion Ratio“ von 35.1dB. Dies stellt eine Verbesserung um 5-10dB zu aktuell verwendeten Methoden dar. Die mittlere „Source to Distortion Ratio“ liegt bei -2.4dB.

1 Introduction

1.1 Motivation

In recent years, digital music libraries have rapidly grown due to the significant increase in distribution of musical content, e.g. by online platforms. With this development comes the need for tools to extract descriptive features for individual musical pieces which can be further processed and linked, e.g to build user recommendations systems. Music information retrieval aims in extracting such information from musical content. Singing voice is of particular interest because of its highly informative character. Many methods dealing with singing voice were proposed over the years, especially in the case of separating the voice from other musical sources. Although the separation quality has increased, it has become not better than satisfactory and thus is still subject of active research.

Separated singing voice is of great interest, since it contains not only the lyrics, which by themselves include a very high amount of information, but also influences significantly the mood or the genre of a song. Furthermore, the extracted singing voice can be used in a variety of applications, e.g. automatic lyrics recognition and alignment, singer identification and remixing applications. To directly extract information from a polyphonic recording is a very challenging task. The more sources present in a recording the more likely misinterpretations will happen or, technically speaking, the more „robust“ the algorithm needs to be against interferences. Therefore, methods to separate or at least attenuate sources in general and singing voice in particular are of high interest. Besides the already mentioned benefits in having the separated singing voice, one motivation behind attenuating the singing voice, from a commercial standpoint is, to build karaoke versions of already established musical recordings without having to re-record the whole piece, which could either be very expensive or might even not be possible, for example, if the artist deceased or the previously used equipment is no longer available.

1.2 Objective

Singing voice separation can have mainly to objectives. First, to leave the accompaniment intact while removing the singing voice (e.g. karaoke applications) and second, to preserve the singing voice and to remove the accompaniment. The latter is the subject of this thesis. Theoretically, if one is able to extract solely the singing voice, one could simply subtract the singing voice from the original recording to gain the accompaniment signal, which we consider a possible additional benefit.

The goal of this thesis is to investigate and implement a suitable algorithm for the task of singing voice extraction using MATLAB®. The input signal is considered to be a 2 channel polyphonic audio-recording in 16bit resolution and sampled at 44.1kHz. Furthermore, we assume the recording to be western music.

1.3 Singing Voice Characteristics

In order to cope with the challenging tasks involved in extracting the singing voice, let us first revisit the main properties of singing voice.

Sounds produced by the human voice can be considered to consist of two parts, voiced sounds and unvoiced sounds. Voiced sounds consist of narrowband sinusoids and unvoiced sounds of broadband or stochastic components. Speech and singing voice share this property, although there are distinct differences, as shown in table 1. Due to the fact that singers intentionally stretch the voiced parts to match accompanying instruments, the singing voice exhibits significantly less unvoiced sounds than speech. For the voiced part, the difference between speech and singing voice lies in the pitch range and the pitch evolution over time. For speech the pitch range is smaller, pitch evolves slower over time and usually drifts down towards the end of a sentence. In contrast, singing voice has a wide pitch range and may exhibit rapid pitch changes.

Property	Speech	Singing Voice
pitch range	80 - 400Hz	80 (bass) - 1400Hz (soprano)
pitch evolution over time	slow drift down, smooth changes	piecewise constant, abrupt changes in between
voiced sounds	~60%	~90%
unvoiced sounds	~40%	~10%
interferences	mostly uncorrelated to target speech	mainly harmonic, broadband and correlated with singing

Table 1: Properties of singing voice compared to speech, from [LW07]

Voiced singing voice

The presence of a fundamental frequency f_0, usually referred to as pitch, and moreover its evolution over time differs between singing voice and instruments. For instance, singing voice exhibits a higher f_0 variability then other musical sources, which makes this a very important property to use in the classification process [Rah09]. Another important property of singing voice is the occurrence of vibrato (Frequency-modulation) and tremolo (Amplitude-modulation) as shown in figure 1. The average vibrato frequency is around 6 Hz [RP09] and the extent ranges between 0.3-1 semitones [KD06]. Additionally, since a singer is not able to produce an f_0 of arbitrary frequency (compare figure 2), algorithms dealing with vocal pitch estimation [Rah09], Besides the existence of f_0 and its proper-

ties, the spectrum of singing voice is of importance. First, it can be assumed to be to harmonic, i.e. each partial frequency is a integer multiple of f_0 [KD06]. Second, singing voice may exhibit accentuated frequency regions called the singer's formant. Figure 3 depicts that this formant can be observed mainly for operatic singers. The reason being, that the perceived loudness of a singers increases by matching the formant frequency with the fundamental frequency [KD06]. In popular music, this is usually not necessary since they rarely perform acoustically. [RVPK08], [Rao09] restrict possible vocal pitches usually within the range 100Hz and 1kHz.

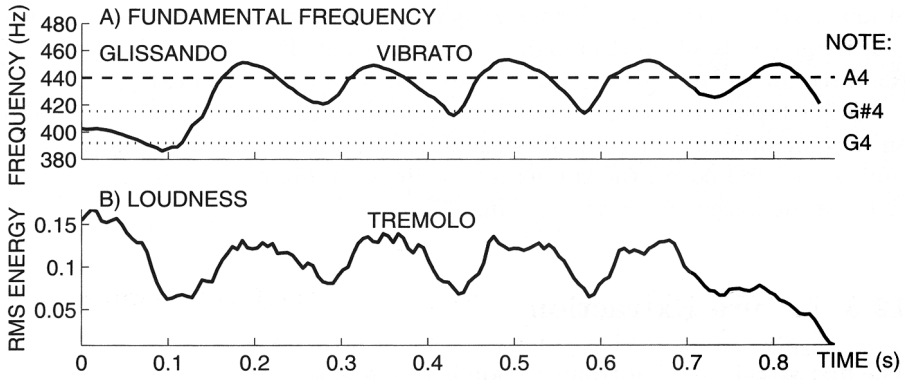


Figure 1: A) The fundamental frequency trajectory and B) the loudness trajectory measured from a note A4 (440 Hz) performed by a female singer. The f_0 curve clearly shows vibrato, whereas the loudness curve shows tremolo. (from [KD06])

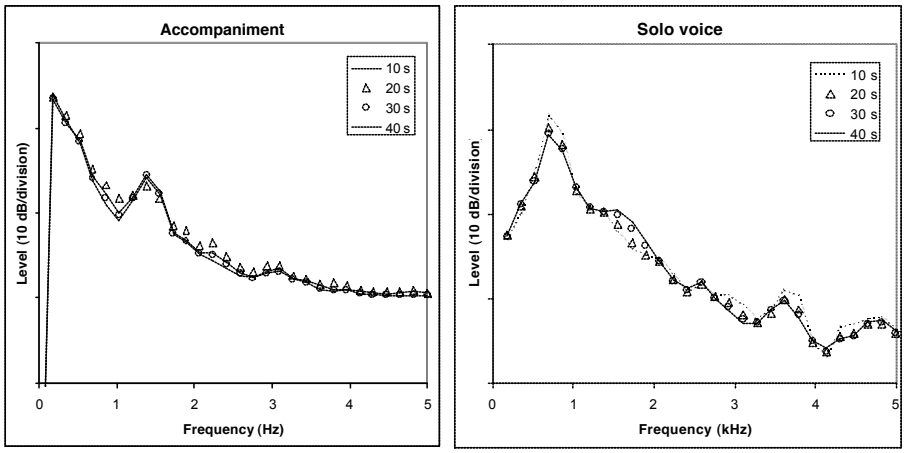


Figure 2: Long-Term-Average Spectrum of singing voice (left figure) and accompaniment (right figure) in pop music for different averaging durations, from [ZBS02]

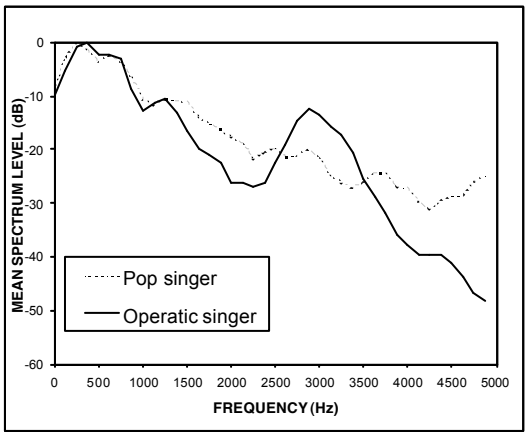


Figure 3: Long-Term-Average Spectrum of a pop singer and a operatic tenor

Unvoiced singing voice

The structure of unvoiced sounds for singing voice and speech can be assumed very similar, if not identical. In the context of this thesis, phonemes consisting of fricatives are of main interest. They usually reside in the frequency range above 2kHz and can exhibit significant energy up to 10kHz [Ter98]. An example of 3 different fricatives and their spectrum is shown in figure 4. In addition, the occurrence of unvoiced sounds is usually very short in duration, i.e. a few 10ms [Ter98].

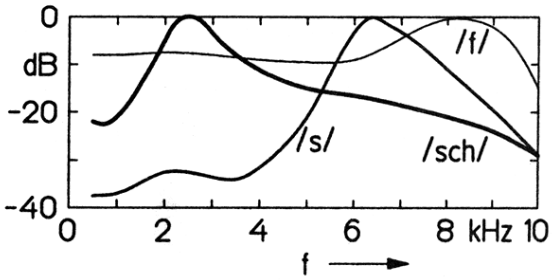


Figure 4: Fourier Intensity Spectrum for 3 different fricatives, from [Ter98]

1.4 Singing Voice Separation

In recent years, many methods in the field of Singing Voice Separation (SVS) have been presented. Although music recordings of last years are predominantly produced in stereo, most of the SVS methods deal with monaural recordings, which is considered to be the more challenging problem. Generally speaking, the underlaying principles can be roughly divided into:

1) **Vectorization or Base Vector Decomposition**
   The goal of this quite common principle is to find all linearly independent components of a signal. Examples are Principle Component Analysis (PCA), Computational Auditory Scene Analysis [Mel91], Non-negative Matrix Factorization (NMF) [CR08] and Independent Component Analysis (ICA) [VB05].

2) **Probabilistic Approaches**
   For instance, Bayesian Models [OPB07], Gaussian Mixture Models [OPGB05], which are usually applied on the STFT to model spectral shapes corresponding to different sources.

3) **Grouping of Segments**
   Initially segments are build, often in the time-frequency domain, which then are classified and grouped to form a mask [CL08], [HJT08], [LW07].

4) **Parametric Approach**
   First a signal model is defined, say a sinusoidal model. Then the parameters of this model are estimated, often introducing a kind of „best-match“ criterion. This approach is used for Vocal Melody Transcription [RVPK08], [Rah09]. The final separation is then based on the estimated vocal melody.

There are of course many methods which do not use solely one distinct approach, but rather a combination. Nevertheless, each approach has its restrictions. For instance a particular problem arises when using NMF which is inability to separate non-stationary sources. Although efforts have been made to extend the model, where an individual non-stationary source is characterized by a set of time-dependent spectral bases (non-negative matrix deconvolution [P.04]), the problem remains challenging and not completely solved. ICA on the other hand requires that the number of sensors (observed mixtures) must be larger or equal to the number of sources. The number of sensors is generally limited to one or two (mono or stereo) channels which is always less than the number of sources, if we assume that singing voice is usually accompanied by more than one other musical source. Probabilistic methods like Bayesian Models require all relevant probability values

to be known, which again is only true very rarely. Signal model driven approaches often suffer under noisy conditions, like reverberant mixtures.

Methods to separate singing voice from stereophonic recordings in particular, often use spatial cues to locate the position of singing voice in the stereo panorama [SAP10], [BL04], [Ave03] or make use of the assumption that singing voice is very likely positioned in the center of the stereo panorama [CL08].

Apart from considering stereo or monaural recordings, extracting the unvoiced component of singing voice has enjoyed little attention [HJT08], mainly because it is regarded to contain a low amount of information, which we belive depends on the intended purpose for extracted singing voice signals.

1.5 Outline

This thesis is organized as follows:

- **Chapter 2**
  describes the proposed method including the differences in processing voiced and unvoiced singing and the use of panning information.
- **Chapter 3**
  explains how the used classifiers are trained and how the used ground truth is build.
- **Chapter 4**
  represents the evaluation of the proposed method
- **Chapter 5**
  discusses the results
- **Chapter 6**
  summarizes this thesis by suggesting possible improvements for future work.

2 Proposed Method

2.1 Overall Structure

We assume that the extraction of the singing voice from a polyphonic mix can be divided into two separate extraction tasks. First, extracting the voiced and second, extracting the unvoiced component of singing voice. The former is solved by locating the appropriate vocal pitches evolving over time and re-synthesizing them and the latter is realized by calculating a time-frequency mask. As shown in figure 5, as a first step, we incorporate the panning information of a stereo recording by assuming that singing voice is very likely to be panned around the center. Next, the voiced and the unvoiced components of singing voice are processed separately and finally the resulting time domain signals are summed up to form the extracted singing voice.

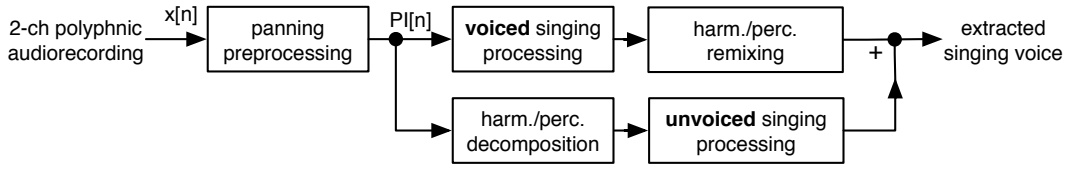


Figure 5: Overall structure of the proposed method

2.2 Panning Preprocessing

Since this thesis deals with polyphonic stereo signals, the panning of individual sources represents an important information which can be used in the task of singing voice extraction. In a multi-track environment, different tracks are mixed using amplitude panning to create a stereophonic effect. Normally, the lead vocal tracks are panned to the center, hence this processing stage tries to benefit from this assumption.
Methods dealing with stereo input signals often initially compute some sort of mono signal in order to reduce the amount of information that needs to be processed. In most cases, both channels are simply summed up, such that frequencies present in both channels are reinforced, assuming they are in phase. In our case, we would like to preserve all frequencies coming from the center, while attenuating those coming from either side with more precession and/or flexibility.
A method to derive such a signal based on the panning is described by Cobos and Lopez [CL08] and consists of the following steps.

If we assume a simplified signal model, with N sources $s_1(t), ..., s_N(t)$, than the left $x_1(t)$ and the right channel $x_2(t)$ can be formulated as follows:

$$x_l(t) = \sum_{j=1}^{N} a_{lj} s_j(t), \qquad l = 1, 2, \tag{1}$$

where a_{lj} are the mixing coefficients which, in most of the Audio Workstations, follow the energy preserving law:

$$a_{1j} = \cos\left(\frac{\Phi\pi}{2}\right) \tag{2}$$

$$a_{2j} = \sin\left(\frac{\Phi\pi}{2}\right) \tag{3}$$

$$a_{1j}^2 + a_{2j}^2 = 1, \tag{4}$$

where Φ is the stereo panning knob value.

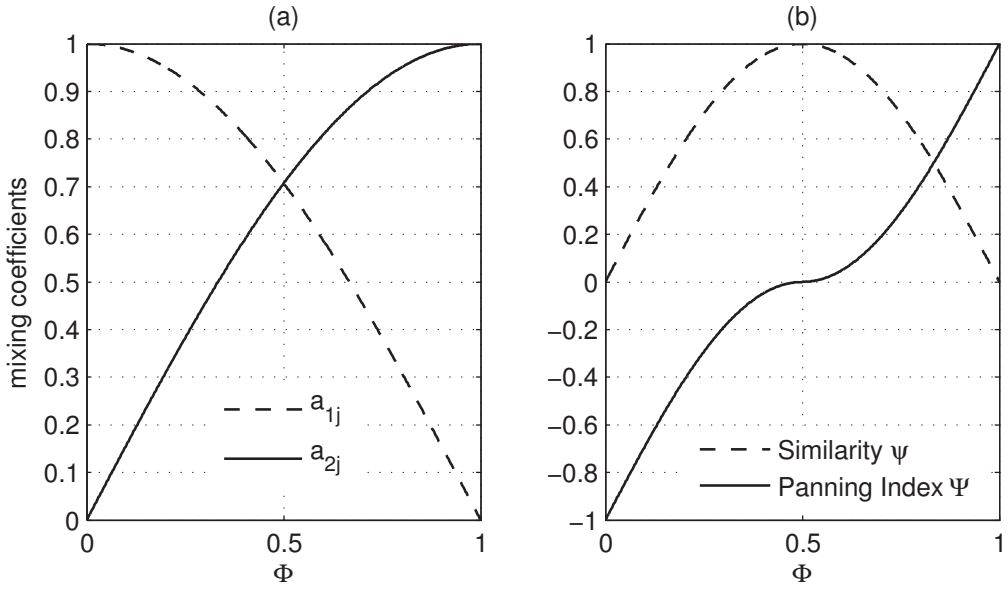


Figure 6: (a) mixing coefficients versus panning knob Φ, (b) similarity and panning index versus panning knob Φ

The relationship between mixing coefficients and panning knob can be seen in figure 6a. Given the linearity of the Short Time Fourier Transform (STFT), the model in Equation (1) can be rewritten such that,

$$X_l[k,m] = \sum_{j=1}^{N} a_{lj} S_j[k,m], \qquad l = 1,2, \tag{5}$$

where k is the frequency index, m being the time index and $X_l[k,m]$ and $S_j[k,m]$ are, respectively, the STFT of $x_l(t)$ and $s_j(t)$.

Next, a similarity measure is defined

$$\psi[k,m] = 2\frac{|X_1[k,m]X_2^*[k,m]|}{|X_1[k,m]|^2 + |X_2[k,m]|^2}, \tag{6}$$

where * denotes the complex conjugation. The function reaches its minimum of zero if the source is panned completely to either side, whereas if the source is panned to the center, the function will attain its maximum value of one. Because of the quadratic components, an ambiguity in knowing the lateral direction of the source is introduced. To resolve this, partial similarity measures are calculated

$$\psi_i[k,m] = \frac{|X_i[k,m]X_j^*[k,m]|}{|X_i[k,m]|^2}, \qquad i \neq j \tag{7}$$

and their difference

$$\Delta[k,m] = \psi_1[k,m] - \psi_2[k,m] \tag{8}$$

which is then expanded to form the ambiguity-resolving function

$$\hat{\Delta}[k,m] = \begin{cases} 1, & \Delta[k,m] > 0 \\ 0, & \Delta[k,m] = 0 \\ -1, & \Delta[k,m] < 0 \end{cases} \tag{9}$$

The panning index $\Psi[k,m]$ finally results in

$$\Psi[k,m] = [1 - \psi[k,m]]\hat{\Delta}[k,m], \tag{10}$$

which identifies the panning locations of all time-frequency components. Theoretically one could simply pick all components for $\Psi = \Psi_0$, but this would have two main drawbacks. First, due to possible interference of different sources and their corresponding partials, the desired source may have significant energy for $\Psi \neq 0$. Second, extracting

only components at Ψ_0 would very likely lead to „musical noise[1]“.

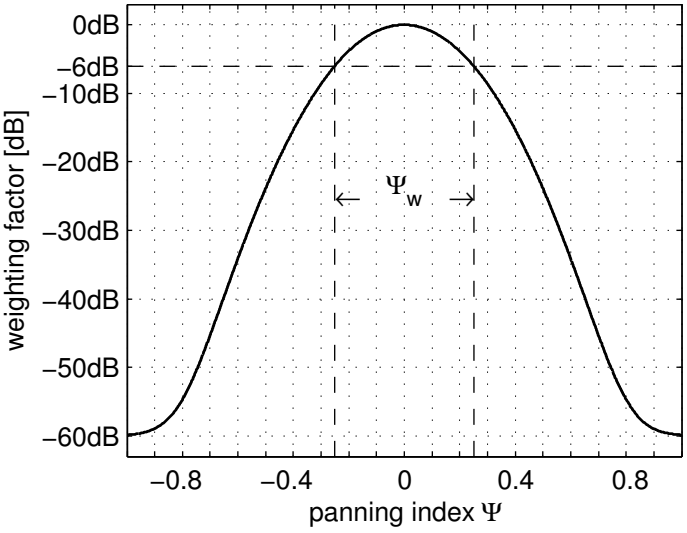


Figure 7: weighting factor versus Panning Index value, window width $\Psi_w = 0.5$

Thus a Gaussian Window $\Theta[k,m]$ (Figure 7) is applied to let components with Ψ_0 pass unmodified and weight components near Ψ_0.

$$\Theta[k,m] = \upsilon + (1-\upsilon)e^{\frac{-(\Psi[k,m]-\Psi_0)^2}{2\zeta}}, \tag{11}$$

where υ is a floor value to prevent musical noise and ζ controls the window width. It can be calculated as follows:

$$\zeta = -\frac{(\Psi_c - \Psi_0)^2}{2\log A}, \tag{12}$$

where Ψ_c is the panning index value where the window reaches the gain value A. For future reference, we define the window width Ψ_w to be

$$\Psi_w = 2 \cdot \Psi_c \tag{13}$$

The specific values were set to $\Psi_w = 0.5$ and $A = -6$ dB leading to the gaussian window shown in figure 7. For a detailed evaluation of the gaussian window width Ψ_w the reader is referred to chapter 4.1 on page 52.

Next, the individual panning filtered signals can be computed by

$$x_l'[n] = IFFT\left\{X_l[k,m]\Theta(\Psi[k,m])\right\} \qquad l = 1,2; k = 1,..,N; m = 1,..,M \tag{14}$$

1. The term „musical noise“ describes the randomly fluctuating spectral components

for every frequency k and every frame m, where N is the FFT length and M the total amount of frames. The filtered left and right channel can then be summed up to form the resulting signal used for the subsequent analysis

$$PI[n] = \frac{1}{2}(x_1'[n] + x_2'[n]) \tag{15}$$

The STFT settings used for the panning index preprocessing are a 4096-point (∼92ms) Hann window with a hopsize of 1024 samples (∼23ms). Figure 2.2 gives an example for the processed panning index values and the resulting weighting factors for a pop music track.

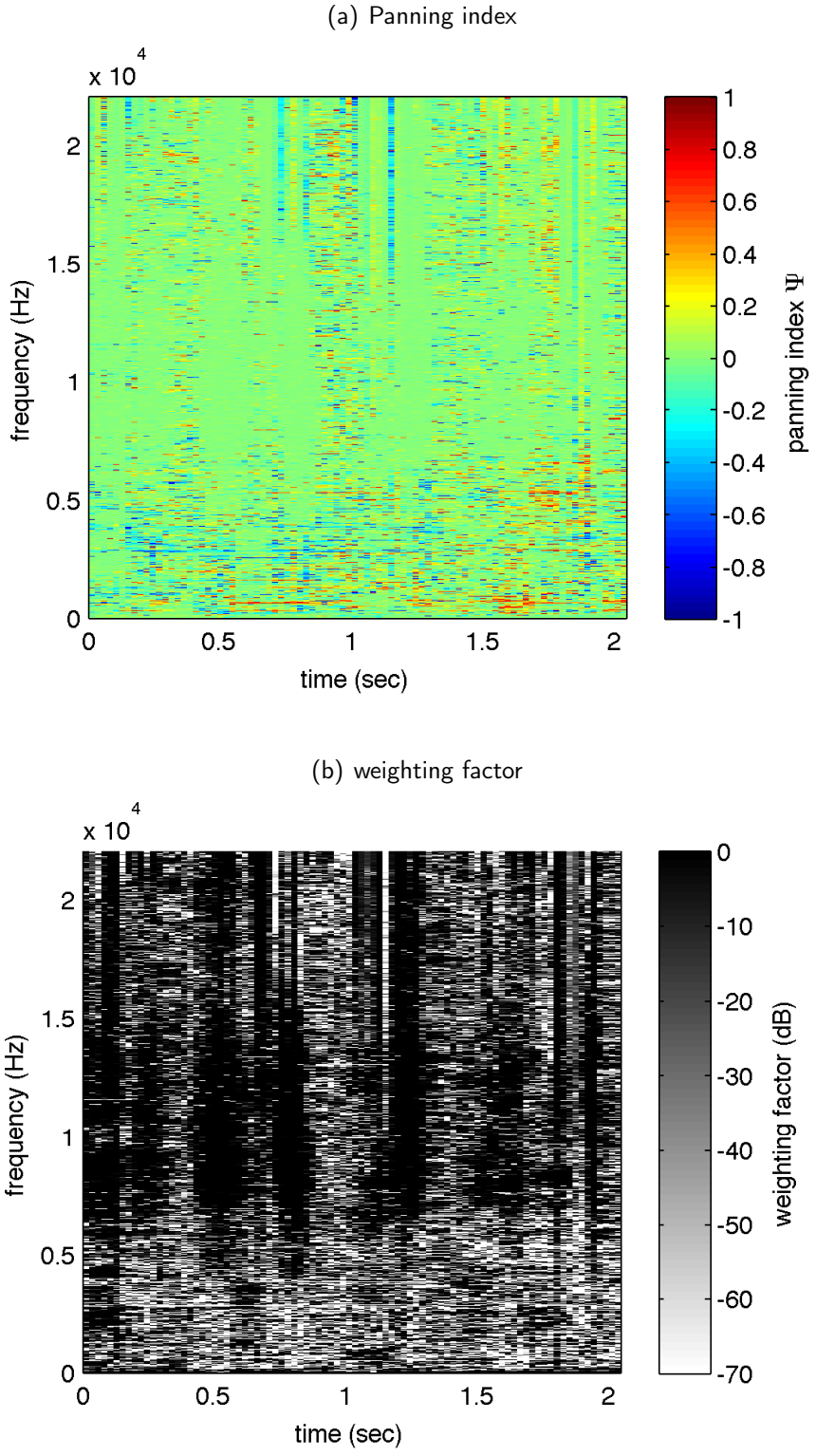


Figure 8: Panning Index example, (a) shows the computed panning index values for each frequency bin and (b) the resulting weighting factors for a pop music track (@44.1kHz, 16bit, stereo)

2.3 Voiced Singing

The extraction of the voiced singing voice is based on the Diploma thesis of A. Rahimzadeh [Rah09], where the author presents the idea to detect and track the vocal pitches, i.e. f_0, using an auditory motivated approach. Since the results were quite promising, we incorporated the key principles of his work along with further improvements. An overview of the proposed method can be seen in figure 9. After processing the stereo input signal with the presented panning index preprocessing, the resulting spectrum is now passed through the auditory processing stage, modeling principles of the human pitch perception. Next, multiple pitch estimates are extracted and their evolution over time is tracked. The decision whether a pitch track belongs to the lead vocals or not takes place in the classification stage, incorporating a Support Vector Machine (SVM). Finally, the time-varying amplitudes, frequencies and phases of the vocal f_0 trajectory along with its corresponding partials are estimated from the original input spectrum and synthesized using Spectral Modeling Synthesis (SMS) (adapted from previous work [Rie09]) to form the voiced singing time domain signal.

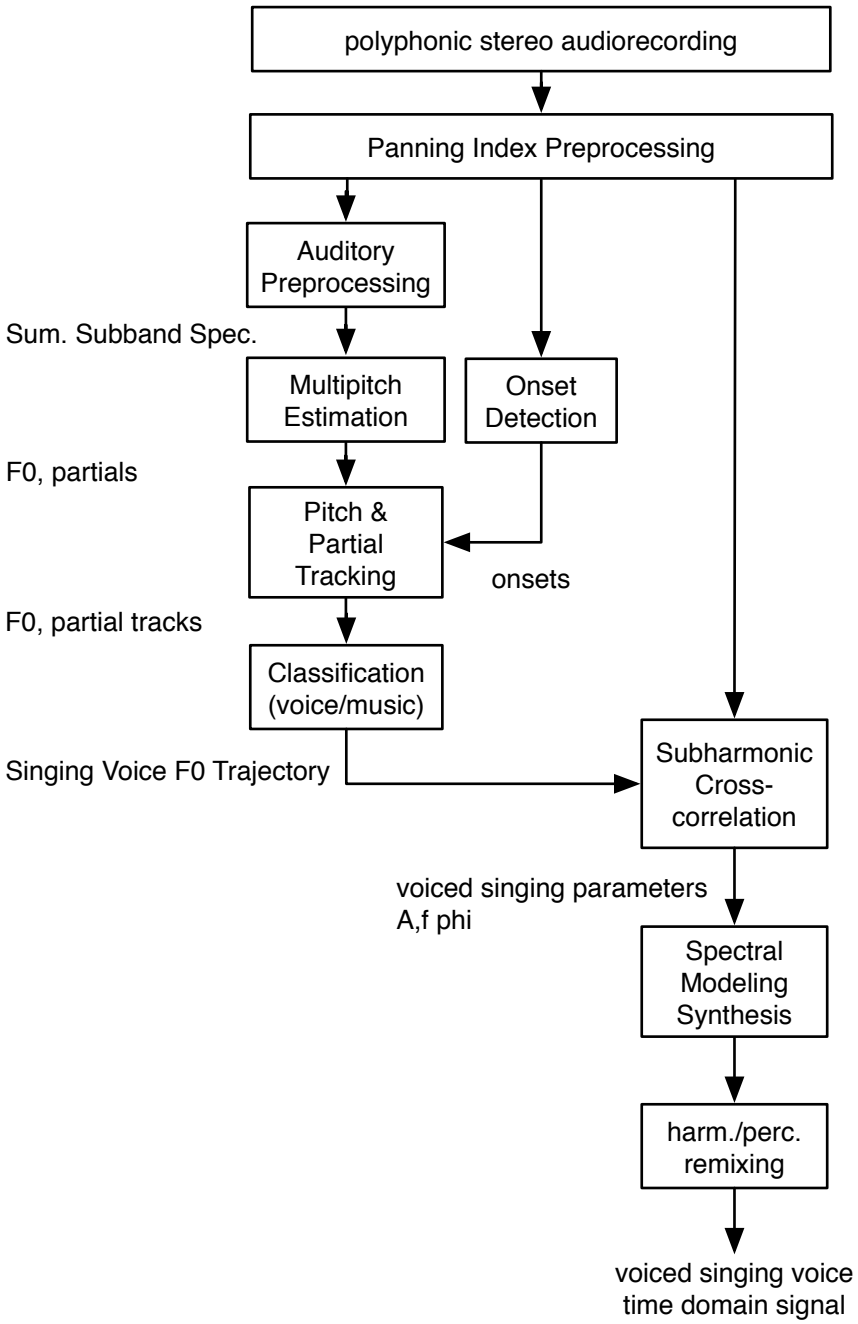


Figure 9: Voiced singing processing stages

2.3.1 Auditory Preprocessing

The vocal f_0 estimation using an auditory motivated approach has been used and studied in detail by Rahimzadeh. Hence, we present a short summary of the essential principles, for detailed informations the reader is referred to [Rah09], [Kla08].

Generally speaking, the auditory preprocessing intends to model the human pitch perception. The proposed computational model [Rah09] that tries to mimic this behavior incorporates the stages shown in figure 10.

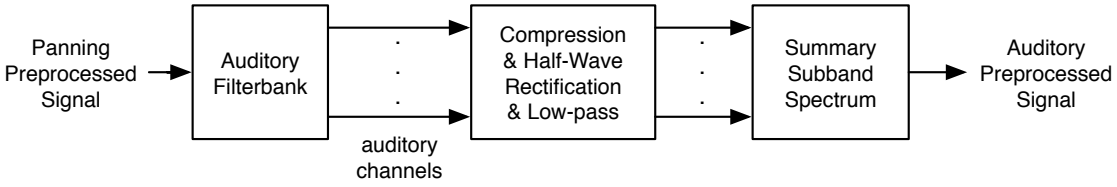


Figure 10: Auditory Preprocessing stages

First, the input signal, in our case the panning preprocessed signal, is passed through a bank of bandpass filters which model the frequency selectivity of the inner ear. These filters are called auditory filters and are realized as gamma-tone filters. A total of 70 filters are used, with center frequencies ranging from 65Hz up to 5.2kHz. These center frequencies are logarithmically spaced, i.e. uniformly distributed on a critical-band scale. The magnitude response for every 3rd filter is shown in figure 11. We use the same filter-implementation proposed by [Rah09], which consists of a cascade of 4 second-order infinite impulse response (IIR) filters.

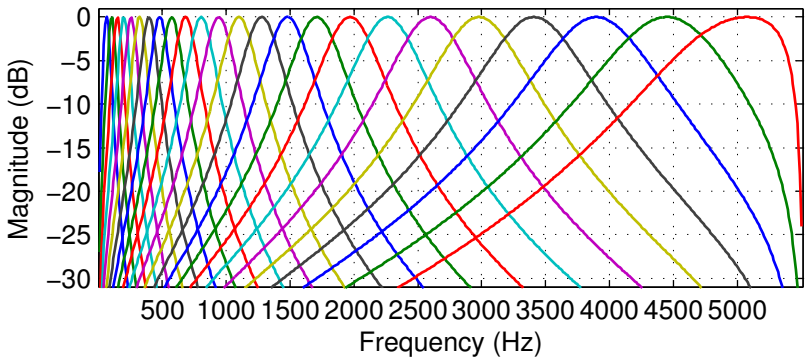


Figure 11: Magnitude response of gamma-tone filters used in Auditory Preprocessing, every 3rd filter is displayed for better readability

To model the inner hair cells and their contribution to the auditory nerve, each filter output (called auditory channel) is now processed separately. First by compression, then

halve-wave rectification (HWR), and lastly low-pass filtering. The latter is performed to suppress frequency components at twice the auditory channel center frequency induced by HWR. Finally, all channels are summed up to build what Rahimzadeh calls the Summary Subband Spectrum, which is the basis for the subsequent analysis.

Apart from improvement in using the auditory preprocessing in the task of pitch estimation, which was presented by [Rah09] and [Kla08], we would like to emphasize one important property. As figure 12 illustrates, the HWR has the effect of generating spectral components not only at multiples of the channel center frequency, but also in the base band. These arise from beating components which corresponds to the frequency intervals between partials. For a strictly harmonic tone complex, this frequency interval is constant and corresponds to the f_0. Even if the tone complex is not strictly harmonic, the most prominent interval will usually correspond to the f_0. Thus, if a harmonic source lacks in f_0 for some reasons, after auditory preprocessing it will be detectable by the pitch estimation stage.

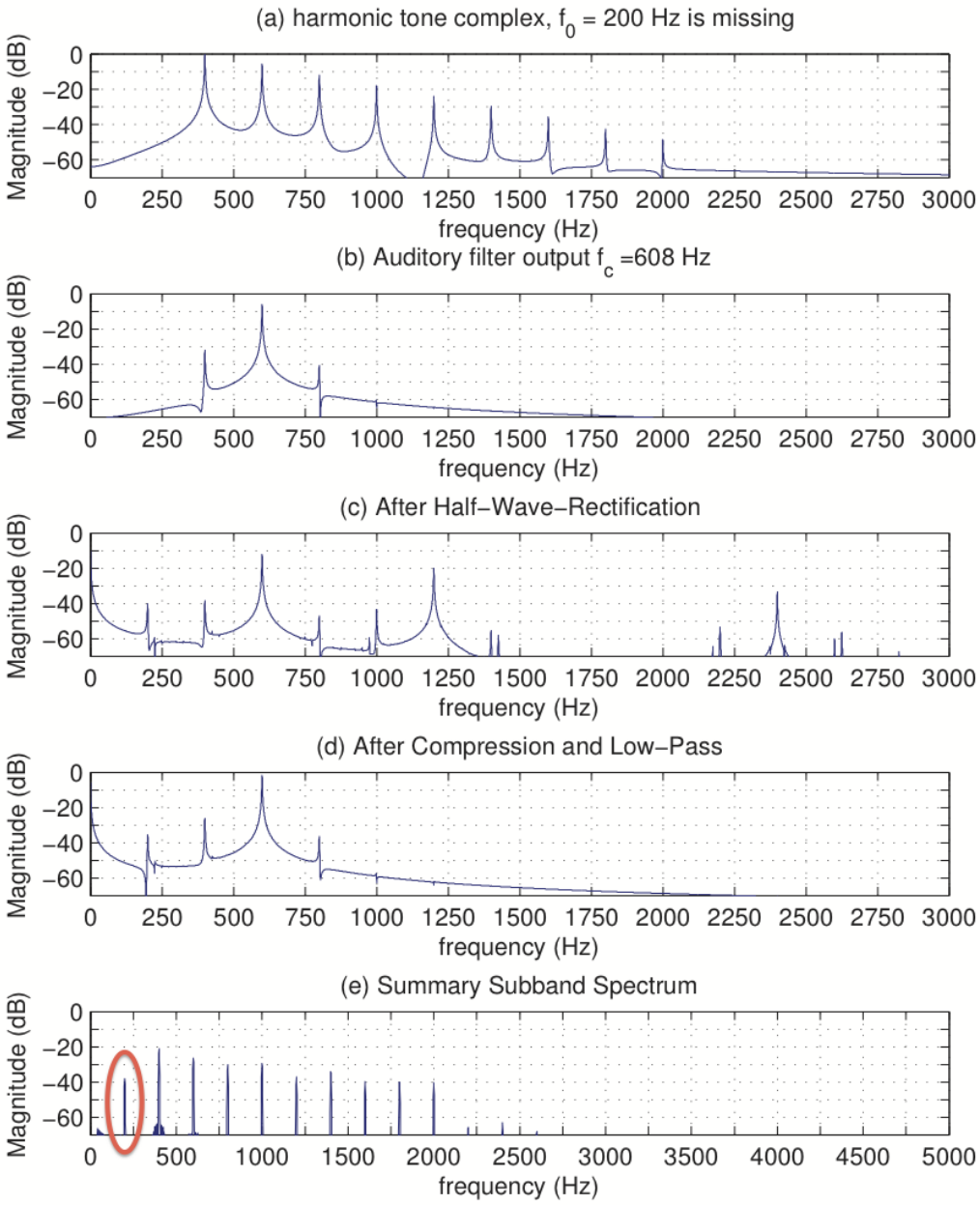


Figure 12: Auditory Preprocessing example for an artificial harmonic tone complex with missing $f_0 = 200$ Hz (red circle)

2.3.2 Multi Pitch Estimation

The Multi Pitch Estimation (MPE) stage performs the detection of the most prominent pitches. In this process, all estimated pitches are initially considered to be pitch candidates, which are then reduced by specific criteria to build the final pitch estimates. Since vocal melodies are restricted by pitches a singer is able to produce (see chapter 1.3), the MPE is performed within a restricted frequency range of 100Hz (∼G#) to 800Hz (∼g2), in accordance to [Rah09]. In order to detect the frame-wise pitch candidates, the following steps are performed. To provide better readability the time (frame) index is omitted.

Firstly, peaks in the summary sub-band magnitude spectrum X_s are detected by taking its 1st order difference

$$X'_s[k] = X_s[k+1] - X_s[k] \tag{16}$$

where k is the frequency index. By evaluating sign changes, a peak is considered detected if

$$X'_s[k] > 0 \quad \text{and} \quad X'_s[k+1] < 0 \tag{17}$$

Next, all peak frequencies along with their corresponding amplitudes are refined using parabolic interpolation (Fig. 13). This is done in using the information of the neighboring bins next to a detected peak.

In using the frequencies

$$\begin{aligned} x_1 = f_1 &= (k-1) \quad \cdot f_s/N \\ x_2 = f_2 &= k \qquad\quad \cdot f_s/N \\ x_3 = f_3 &= (k+1) \quad \cdot f_s/N \end{aligned} \tag{18}$$

and magnitudes

$$\begin{aligned} y_1 &= 20log_{10}|X[k-1]| \\ y_2 &= 20log_{10}|X[k]| \\ y_3 &= 20log_{10}|X[k+1]| \end{aligned} \tag{19}$$

as well as the parameters α and γ

$$\alpha = \frac{y_2 - y_1}{(x_2 - \gamma)^2 - (x_1 - \gamma)^2} \tag{20}$$

$$\gamma = \frac{1}{2} \cdot \frac{(y_3 - y_1)(x_2 - x_1)(x_2 + x_1) - (y_2 - y_1)(x_3 - x_1)(x_3 + x_1)}{(y_3 - y_1)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_1)} \tag{21}$$

originating from the parabolic function

$$y(x) = \alpha(x - \gamma)^2 + \beta \tag{22}$$

the refined magnitude $X'_{mag}[k]$ can be calculated by

$$X'_{mag}[k] = y_2 - \alpha(x_2 - \gamma)^2 \tag{23}$$

while the refined frequency is represented by γ. All candidates with frequencies outside the restricted vocal f_0 range are discarded.

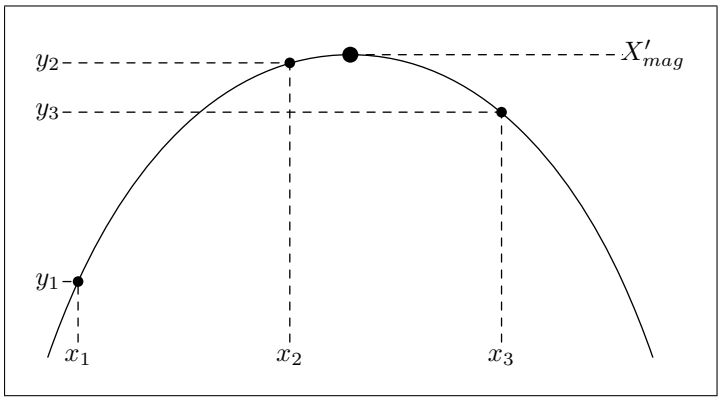


Figure 13: parabolic interpolation

Furthermore, we restrict the total amount of pitch estimates per frame to 10. Hence, if more estimates were found those with highest amplitudes are kept. All remaining candidates are considered to be valid pitch estimates and passed on to the pitch tracking stage.

The pitch estimator introduced here represents a peak detection in the FFT domain. Hence, it is important to mention its frequency discriminability. It depends on the FFT window length and shape in addition to the sampling frequency. As already mentioned, the pitch estimation is based on the summary sub-band spectrum X_s, which is obtained using a $N = 4096$ point Hann window at $fs = 44.1$ kHz sampling rate. This results in a frequency resolution of fs/N = 10.77Hz. The Hann window itself has a main lobe width of 2 bins, which degrades the resolution by 2 to 21.53 Hz. This means that we are able to resolve a interval of 2 semitones (≈12%) for frequencies above ∼180Hz (≈Note f). This seems a reasonable frequency resolution, since in popular music, intervals less

than 2 semitones are rarely played together, as they would result in perceptually harsh sounds, at least in this low frequency range.

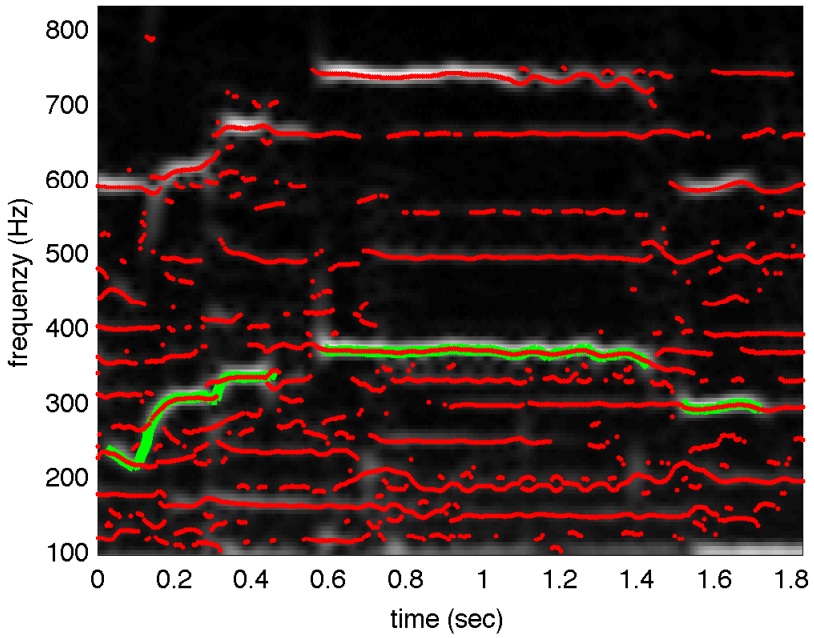


Figure 14: Pitch Estimation Candidates (red), reference f0 (green), 10 candidates per frame

2.3.3 Pitch and Partial Tracking

The frame-wise estimated pitch candidates are now grouped together to form continuous pitch tracks. This stage is based on an adapted previous work. For a detailed description the reader is referred to [Rie09]. Basically, for each frame, 3 different cases can be distinguished, depending on the existence of pitch candidates and pitch tracks:

a) existing pitch candidates, but non-existing pitch tracks

This is the case if pitch candidates are detected for the first time or if all pitch tracks were terminated in the previous frame. All pitch candidates are sorted by their frequency in ascending order and processed one by one. In a first step, a pitch candidate p is selected to form a new pitch track. Next, the frequency range Δf_p is computed, in which other pitch candidates are considered to be possible candidates for the pitch track under investigation.

$$\Delta f_p = \Delta_f \cdot f_p \quad \text{with} \quad 0 \leqslant \Delta_f \leqslant 1 \tag{24}$$

where f_p is the frequency of pitch candidate p. If multiple candidates were found, the one with highest amplitude is selected, while the others are excluded from the

subsequent analysis. Δ_f is empirically set to 0.045 ($\widehat{=}$ 77 cents) and represents not only the mentioned frequency range for possible candidates, but also the maximum frequency deviation for a pitch track for successive frames.

b) existing pitch candidates and existing pitch tracks

First, existing pitch tracks are considered. Since the pitch track frequency will evolve over time, the above-mentioned maximum frequency deviation Δ_f (Eq. 24) is calculated for each track. In this range, we try lo locate other pitch candidates and, if found, the one with highest amplitude is selected. In case that two pitch tracks have overlapping Δ_f regions, the selected pitch candidate is assigned to the track which has the highest mean amplitude since birth. This is done to attenuate the importance of tracks with low mean amplitude. Every other pitch estimate located within this range is excluded from subsequent procedures. If no estimate was found, the pitch track under test will be terminated. After all pitch tracks have been updated, remaining unassigned pitch candidates are processed in the manner described in a).

c) non-existing pitch candidates, but existing pitch tracks

All existing pitch tracks are terminated.

The pitch tracking process is based on amplitude considerations (Figure 15), i.e a pitch track follows the most prominent magnitude in its frequency neighborhood. This may induce the following problem. If an instrument has a strong continuing spectral component overlapping in frequency with a vocal pitch track, this track is continued, regardless if the singer is still singing or not. Building pitch tracks containing both unwanted sources and the desired source should be avoided or at least minimized, since this would decrease the reliability of the resulting feature values. As a result those tracks would not allow a distinct classification. We assume this problem is more likely to occur at time instances related to the rhythmical structure of a musical piece, therefore propose a onset detection based on the spectral flux such that, on each onset every pitch track is split. The resulting pitch tracks will then be classified separately.

The Spectral Flux measures the change in magnitude in each frequency bin and is defined by

$$SF[n] = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H[|PI[n,k]| - |PI[n-1,k]|] \tag{25}$$

where N is the window length, $PI[n]$ being the panning index preprocessed signal at

time instant n and $H[x]$ representing the rectifier function defined by

$$H[x] = \frac{x + |x|}{2} \tag{26}$$

Additionally, occurring onsets are restricted to have a minimum spacing of 400 ms (500ms $\widehat{=}$ 1/4 @ 120 BPM), to avoid that reliable tracks are split and then discarded due to the Minimum Life Time (MLT) constraint. This constraint states, that pitch tracks shorter than 50ms (60ms $\widehat{=}$ 1/32 @ 120BPM) are discarded. Furthermore, pitch tracks are also discarded if they have a low f_0 salience. The f_0 salience is calculated as the mean f_0 amplitude for each pitch track. It is compared to the local mean salience, which is the mean f_0 salience for tracks surrounding the track under test in a $\pm$ 1 sec window. If the track has less than 50% of the local mean, it is discarded. The MLT and f_0 salience methods to increase pitch track reliability were described and introduced by Rahimzadeh [Rah09], which we implemented accordingly.

To conclude, figure 15(b) gives an example of the presented post-processing which in this case deleted $\sim$63% of all tracks. Additionally, figure 15(b) illustrates the improvement in using onset based track splitting to separate unreliable parts of vocal pitch tracks.

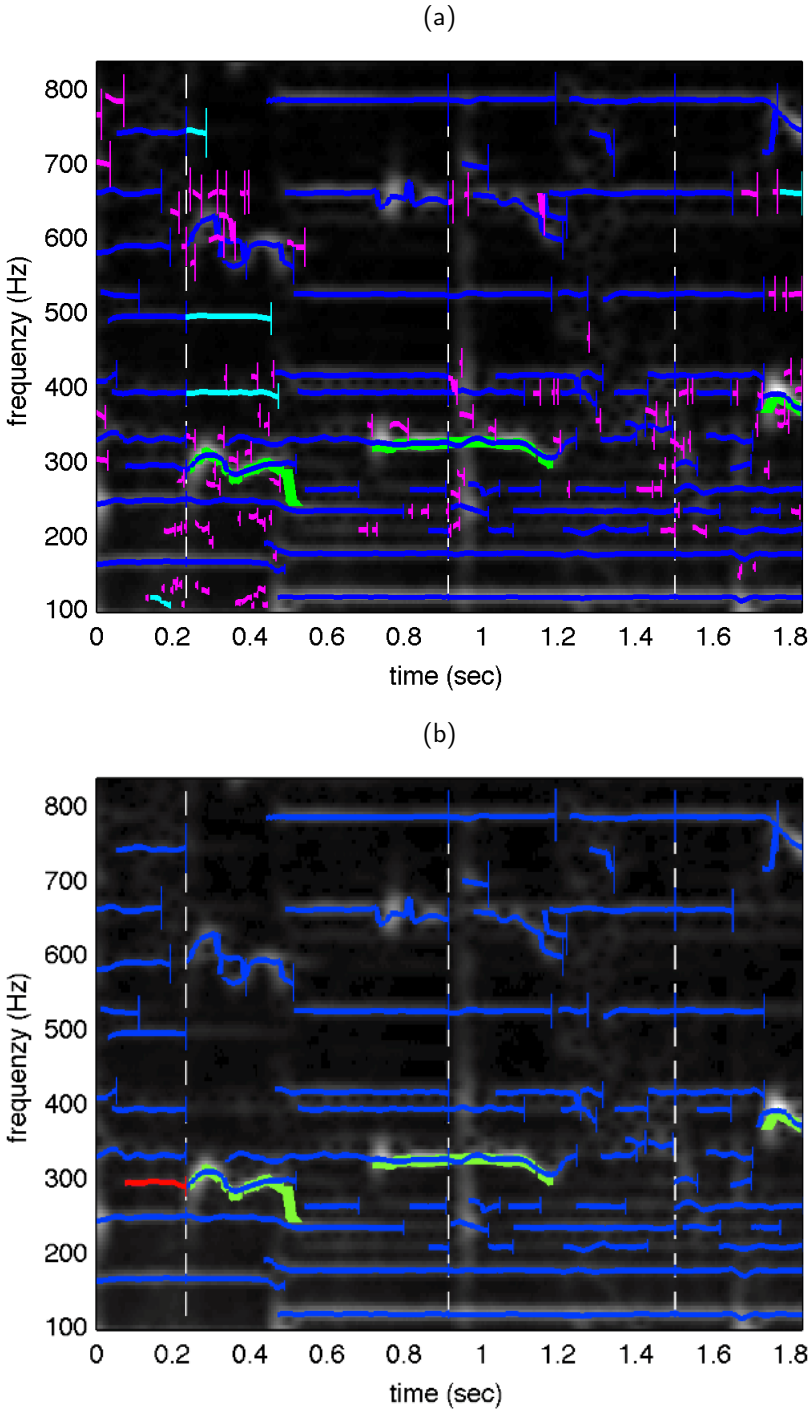


Figure 15: Example of pitch tracks and post-processing, (dashed white line) detected onsets, every other vertical line marks the end of a pitch track, (green) reference f_0
(a)(blue) 190 tracks build, (magenta) ∼59% of tracks deleted because of Minimum Life Time, (cyan) ∼3% of tracks deleted because of low f_0 salience
(b) remaining pitch tracks after post-processing, (red) unreliable part of track which could be separated using onset based track splitting

2.3.4 Pitch Track Classification

The task of the classification stage is to classify pitch tracks into the classes *vocal* or *music*. There are many different classifiers currently used in the broad field of classification. One of the most powerful, when it comes to precision and performance, is the Support Vector Machine (SVM), although training the model and finding the best parameters can be challenging. The SVM used in this work, was compiled using the online available LIBSVM source files provided by Chang, Lin [CL].

Due to the promising results achieved by Rahimzadeh, we incorporate his feature set consisting of 27 features (for a detailed description the reader is referred to [Rah09](page 45-48)):

1) Absolute strength of harmonic series:
the sum of all partial amplitudes averaged over the track length

2) Mean Relative Salience:
ratio of spectral energy of partial tone series and remaining energy in the frequency range of 300Hz - 2.5kHz

3) Mean f_0

4) Summary partial standard deviation

5-10) Partial standard deviation:
standard deviation of every partial frequency trajectory

11-13) Summary delta-partial tone frequency:
the sum of absolute frequency difference between consecutive frames for every partial tone, summed up for all partials

14-15) Absolute f_0 range in frequency and amplitude:
difference of maximum and minimum f_0 value

16) Relative f_0 range:
absolute f_0 range on a cent scale

17-20) Δ_{f0}:
Difference in frequency from frame to frame

20) Mean Δ_{f0}

21) Δ_{f0} Standard deviation

22) Δ_{f0} Variance

23) Δ_{f0} Maximum

24-28) Mean partial amplitude ratio:
ratio of mean partial amplitude of pitch track under test and mean partial amplitude of tracks surrounding $\pm$0.5 sec

Additionally we introduce:

29-31) Variance of deviation of partial amplitudes for partials 1-2, 1-3 and for all partials:
first, the increase in amplitude for each partial for successive frames is calculated, finally the variance which results for the partials 1-2, 1-3 and all partials is calculated

32-34) Average amplitude ratio of first 2, first 3 and all partials to the f_0 amplitude

35) Spectral flux (Eq.25)

36-40) f_0 Amplitude increase at the beginning of a pitch track for the first 5 frames

41-42) Average variance and standard deviation of f_0 Amplitude

43-44) Average variance and standard deviation of f_0 Frequency

45) f_0 Vibrato range:
calculated as the difference between the maximum and minimum increase in f_0 frequency, averaged over the pitch track length

46-48) Vibrato frequency [RP09] of first 3 partials

49-51) Tremolo [RP09] of first 3 partials

The complete feature set is decreased by calculating the Fisher's Ratio (FR) (see equation 33 on page 44) and setting an empirically determined threshold to 0.1. In this way, all features that examine a FR of < 0.1 are not included in the final feature subset. Figures illustrating each feature and its corresponding FR are presented in the chapter 3.1 „Training of Classifiers“ on page 47, where as the exact values can be studied in the Appendix A.

Furthermore, a Principal Component Analysis (PCA) is performed. Experimental results showed, that the classification performance significantly increases (5-10%) by using this preprocessing step. For more details the reader is referred to Chapter 4.2.3 on page 63. It should be mentioned, that the PCA is used to transform the feature set onto orthogonal feature space and not to investigate the amount of underlying components. Experiments showed, that discarding components, as well as restricting the total variance covered by them, only decreased the classification performance.

After pitch track classification, a post processing on the vocal pitch tracks is performed. This is necessary, since we allow the SVM to classify multiple pitch tracks as class *voice*,

which can overlap in time. Obviously, only one vocal pitch track can be present per time instant. Hence, we have to decide after the classification process, which vocal pitch track is more likely to be the true one.

Mainly, two different cases can be distinguished for the occurrence of overlapping vocal pitch tracks. First, the overlapping tracks can actually be the 1st and 2nd partial of the singing voice (figure 16). Since they originate from the same source, they exhibit similar feature values. Second, only one track originates from singing voice. As a possible solution to this problem, we propose a rule-based approach. First, we detect the start and end point of the overlapping segment by so called split points M_1 and M_2 (green vertical lines in figure 16). Between those markers, the Summary Mean Spectral Amplitude (SMSA) is calculated for each pitch track. The SMSA is the mean amplitude of the first 4 partials during the overlapping segment and is calculated as follows:

$$SMSA(i) = \frac{1}{M_2 - M_1 + 1} \sum_{p=1}^{4} \sum_{m=M_1}^{M_2} |X(f_{p,i}, m)| \quad \text{with} \quad f_{p,i} = p \cdot f_{0,i}, \tag{27}$$

where $|X[f_{p,i}, m]|$ is the magnitude of partial-frequency $f_{p,i}$ of pitch track i between the frames M_1 and M_2. The pitch track with the highest SMSA is chosen for the intersection (lower red pitch tracks with highlight in core, figure 16). In the case of two singing voice partials overlapping in time, the SMSA proved to be a good decision method during our experiments, since the 2nd partial has usually less energy at its partial frequencies. Even if the second overlapping pitch track originates from an instrument, the SMSA of the true voice pitch track is usually higher. However, to account for the fact that there are instances where the SMSA based approach choses the „wrong“ pitch track, we introduce a second post-processing step.

If, for instance, in an overlapping segment of vocal pitch tracks the 2nd partial was chosen over the 1st, the resulting final voice pitch track would examine a rapid f_0 increase or decrease at the initial overlapping boundaries. Therefore, we locate f_0 increments or decrements by more than 50Hz from frame to frame. At such time instances, the mean f_0 frequency is calculated in a $\pm$0.5 second window. Next, if multiple vocal pitch tracks exist, the one with the least difference in frequency to the calculated mean f_0 is chosen. If only one vocal pitch track exists, the resulting final vocal pitch track is split to avoid rapid frequency sweeps in re-synthesis.

Additionally, we allow the final vocal pitch track to drop out, i.e to be terminated and reborn shortly after, for a predefined time interval called Maximum Rest Time (MRT) of 20ms ($\sim$4 frames) and still be continued (figure 17). In this case, the pitch track

frequency is linearly interpolated to fill the gap.

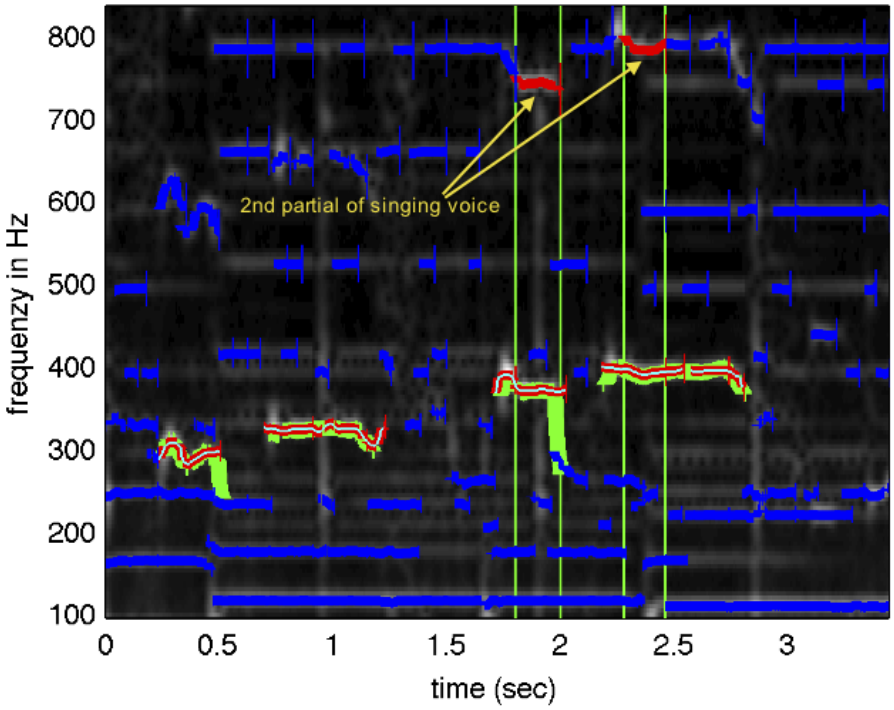


Figure 16: Postprocessing of classification results by Summary Mean Spectral Amplitude (SMSA), (red) pitch tracks of class voice, (blue) pitch tracks of class music, (green) reference vocal pitch track, (green vertical lines) boundaries of overlapping vocal pitch tracks, pitch track with the highest SMSA is chosen for the intersection, (red pitch tracks with highlight in core) final voice pitch tracks after post processing

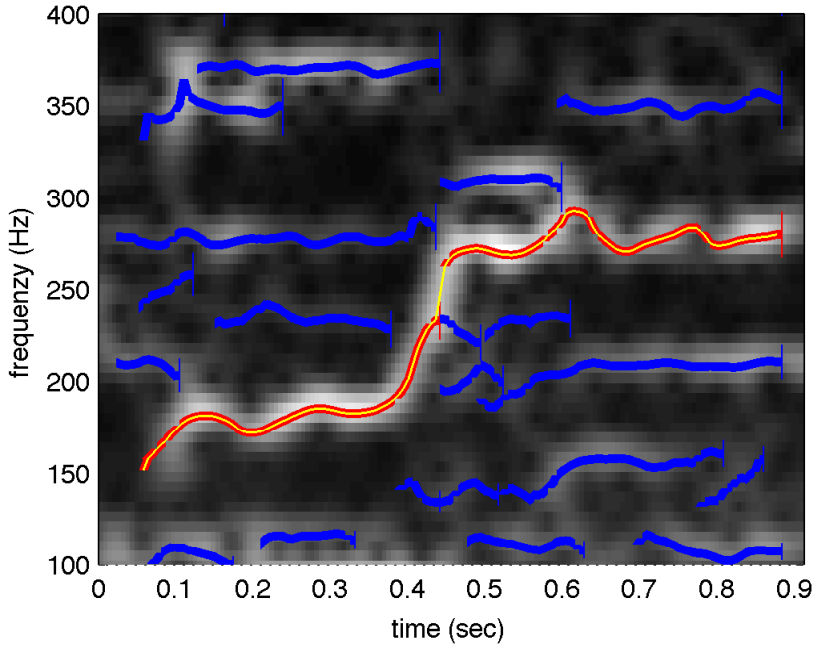


Figure 17: Postprocessing of classification results by Maximum Rest Time (MRT), (red) pitch tracks of class voice, (blue) pitch tracks of class music, (yellow) final voice pitch tracks after post processing

2.3.5 Spectral Parameter Estimation

In order to increase the time resolution, a shorter window (~23ms) and hopsize (5.8ms) is used in comparison to the analysis process. Therefore, the classified vocal frequency trajectory is interpolated using cubic splines. The trajectory values that have to be interpolated are found by matching the temporal positions of the corresponding windows.

Furthermore, to extract the voiced singing voice, not only the f_0 has to be known, but the corresponding harmonics. We use the method proposed by Ryynanen et al. [RVPK08] which consists of a Normalized Cross-Correlation NCC between a complex exponential based on the partial frequency f_p and the analysis frame. It can be calculated by

$$NCC[f_p] = \frac{\sum PI[n] exp(i2\pi n f_p/f_s) w[n]^2}{\sum^n w[n]^2/2} \tag{28}$$

where f_s is the sampling frequency, $PI[n]$ the panning index preprocessed input signal and $w[n]$ a Hamming window, centered at the temporal position of the analysis frame. In order to get the amplitude a_p and phase ϕ_p values for partial p, the magnitude and

phase (angle) respectively is computed by

$$a_p[f_p] = |NCC[f_p]| \quad \text{and} \quad \phi_p[f_p] = \angle(NCC[f_p]) \tag{29}$$

It should be mentioned, that the partial frequency is restricted to integer multiples of f_0.

2.3.6 Re-Synthesis

This stage is based on previous work, hence only a short summary is presented here. For a detailed description, the reader is referred to [Rie09].

The vocal f_0 trajectory, along with its partials, estimated by the previous stage, is now re-synthesized using Spectral Modeling Synthesis (SMS) [Ser97]. The SMS Model states that a complex signal can be split up into an deterministic part and a stochastic part. The deterministic part consists of single sinusoids, while the stochastic part consists of noise or noise-like broadband components. Here, we focus only on the deterministic part, since the stochastic part is covered by unvoiced singing processing stages. In our case, the sinusoids are the partial frequency trajectories of singing voice.

For the purpose of readability, the following section describes the re-synthesis process for the f_0 trajectory, having in mind that that the steps are repeated for all partial frequencies.

The vocal pitch track can basically match with one of these 3 states:

1) pitch track is born

If the vocal pitch track is born in the actual frame m, it is re-synthesized starting in the previous frame $m-1$. The amplitude consists of a linear fade from 0 to the actual amplitude value in frame m. The detected phase value $\phi_{f_0}[m]$ is used to determine the zero-phase value in the previous frame

$$\phi_t[m-1] = \phi_t[m] - 2\pi \frac{f_0}{f_s} N_{hop} \tag{30}$$

where f_s is the sampling frequency and N_{hop} the hopsize in samples.

2) pitch track is terminated

If the vocal pith track is terminated, i.e. no longer present in the actual frame, it is linearly faded-out in the previous frame.

3) pitch track continues

The values for, frequency and phase in the actual frame and last frame are interpolated using a cubic phase interpolation proposed by McAulay and Quatieri [MQ86]. The amplitude values are linearly interpolated.

Again, the presented method to re-synthesize f_0 is performed for all partial frequencies with a frequency $<$15 kHz . Additionally, since the analysis of the partial trajectories, i.e. their actual amplitude, frequency and phase values, is performed every hopsize seconds, the re-synthesis is also based on the length of one hopsize ($\sim$6ms).

Experimental results showed that synthesizing partials up to 15kHz increases the intelligibility, but also increases the likelihood that energy at such high frequencies belongs to percussive sources. Therefore, the re-synthesized singing voice is decomposed into its percussive and harmonic part and then remixed using a percussive to harmonic ratio of -6dB. The decomposition is an essential part of processing unvoiced singing, thus is explained in detail in the following chapter (page 34).

2.4 Unvoiced Singing

The unvoiced singing stage is based on the principles proposed by [HJT08], which consists of an unvoiced frame detection followed by dividing the signal in time and frequency resulting in so called Time-Frequency-Units, which then are classified and finally extracted. Although, the author presented promising results, the computational costs were very high. Therefore, we decided to decrease the complexity by reducing the necessary amount of information, needed for the unvoiced singing extraction. As a first step, the input signal is decimated by factor of 2, resulting in a sampling frequency of 22.05 kHz.

As shown in figure 18, the process of identifying and extracting the unvoiced singing voice is again based on the panning index preprocessed signal. First a harmonic/percussive decomposition [Fit10] is introduced and only the resulting percussive signal is used for the subsequent processing. Next, this signal is high-pass filtered and passed through a gamma-tone filter-bank and each channel of the filter-bank is split up into frames resulting in Time-Frequency-Units (TFU). Only TFU's from previously detected unvoiced dominant frames are used in the subsequent feature-extraction and binary classification using a SVM. After post-processing, the resulting mask is used to extract the unvoiced singing TFU's.

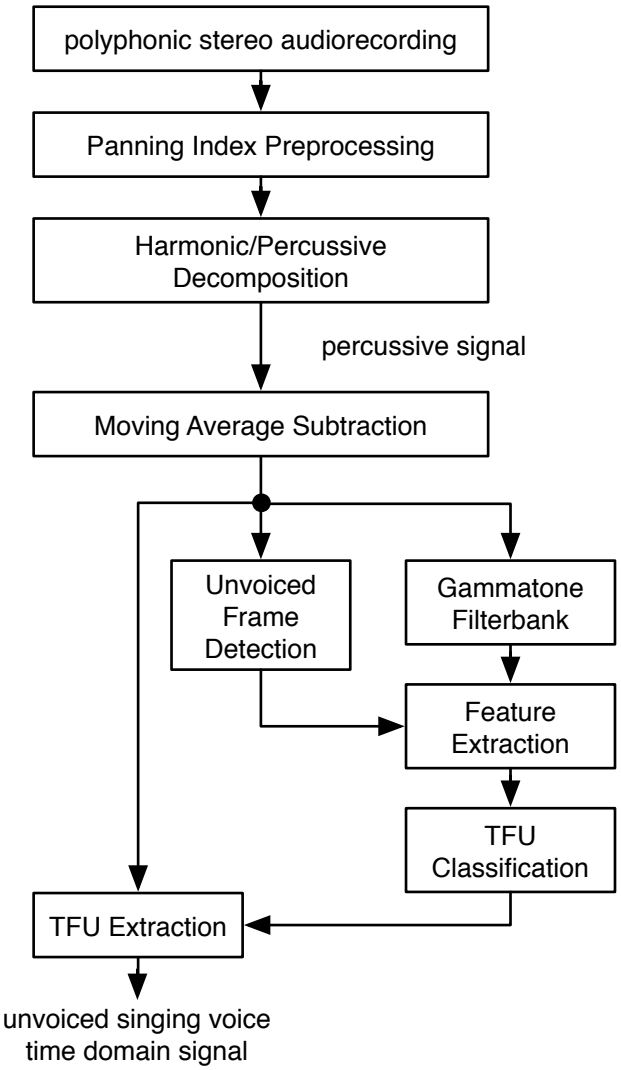


Figure 18: Unvoiced singing processing stages

2.4.1 Harmonic/Percussive Decomposition

To improve the subsequent analysis, the panning index preprocessed signal (see chapter 2.2) is further processed in 2 steps. First, by decomposing the PI signal in a harmonic and percussive part and second by moving average subtraction.

A computational efficient method to perform this decomposition was proposed by FitzGerald [Fit10]. Looking at a spectrogram as in fig 19, the general idea is that percussive events can be identified by strong vertical lines, in contrast to harmonic events which appear as horizontal lines. To separate one from the other, the signal is median filtered two times. First, along each frequency bin and second, for each time instant (frame). The former will result in a signal dominated by harmonic sources, while the latter will preserve percussive events.

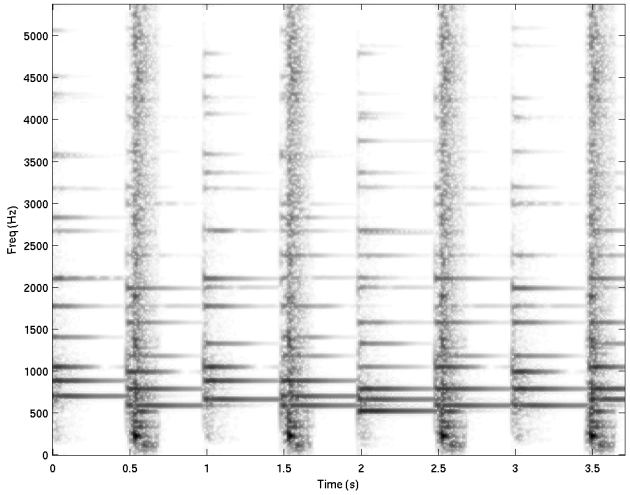


Figure 19: Spectrogram of pitched and percussive mixture, refer to [Fit10]

Experimental tests revealed, that the percussive signal can still contain tonal components. Therefore, the percussive signal is filtered using moving average subtraction. The moving average subtraction can be seen as a high pass filtering, where the filter length is proportional to the resulting cutoff frequency. We implemented a moving average subtraction corresponding to a high pass filter with a cutoff frequency of 2.2 kHz.

For the subsequent analysis only the high-pass filtered percussive signal is considered.

2.4.2 Unvoiced Dominant Frame Detection

The Unvoiced Dominant Frame Detection (UVFD) is performed to decrease the amount of frames considered in the subsequent classification process.

The identification of unvoiced dominant frames is realized by combining two parameters, the presence of singing voice and the variance of the linear prediction error signal. The presence of a singing voice f_0 at a certain time instant is detected by the voiced singing stage. Additionally, to account for the fact that unvoiced singing components are very likely to occur prior and shortly after a singing voice f_0 is detected, an additional time window is introduced, which is referred to as pre- and post-listen respectively. This dependency on the presence of singing voice is introduced for two reasons. First, we assume that unvoiced singing components do not interrupt the detection of the singing voice f_0 trajectory, due to their short time duration in conjunction with the restricted time-resolution of the voiced singing STFT processing. Second, restricting the final unvoiced singing extraction to time instances that correlate with the presence of singing voice will increase the audible quality. The second parameter to identify unvoiced dominant frames, is the exceedance of a threshold on the error signal variance resulting from a 5th order Linear Prediction (LP).

To summarize, if in a specific frame the variance of the LP error signal exceeds a certain threshold and furthermore a singing voice f_0 is present, including the mentioned additional time window, this frame is considered to be unvoiced dominant. An example can be seen in figure 20, where the red rectangle in the top row marks unvoiced dominant frames.

The parameter settings where found empirically (see evaluation, page 68) and set according to table 2.

Parameter	**Value**
LPC error threshold	$10^{-4.25}$
f_0 pre-/post-listen	20 ms

Table 2: Parameter settings for Unvoiced Frame Detection

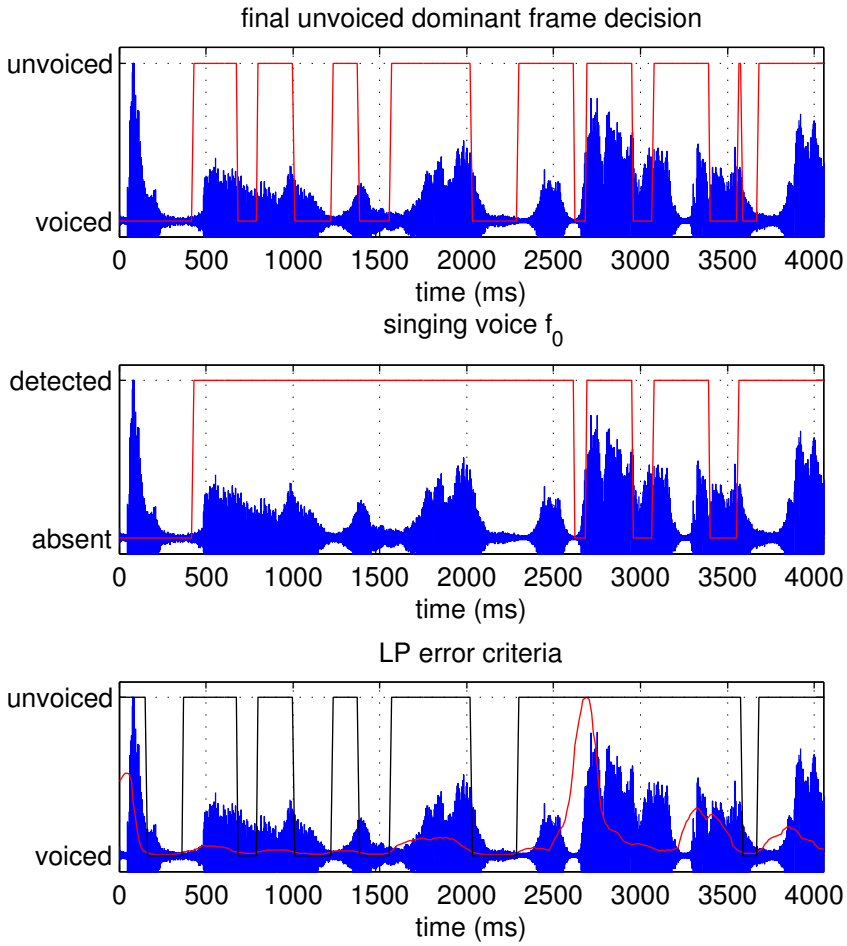


Figure 20: Example for Unvoiced Dominant Frame Detection, (blue) singing voice time domain signal, [top row] final unvoiced frame decision, [middle row] presence of singing voice f_0, [bottom row] (red) Linear prediction error variance and (black) resulting dominance decision

2.4.3 Time-Frequency Units

In the next step, the signal is split up into several frequency bands which are distributed on a critical band scale. A total amount of 7 gamma-tone filters are used with quasi-logarithmically spaced center frequencies ranging from 2kHz to 9kHz, which is roughly the frequency range where unvoiced sounds (fricatives, plosives etc.) can be expected [Ter98]. Finally, each gamma-tone filter output is decomposed into overlapping frames of ∼90ms in length. In that way the input signal is split into segments in frequency and time, resulting in so called Time-Frequency-Units (TFU).

2.4.4 TFU Classification

The previous UVFD separated the unvoiced from the voiced dominant frames. Now, the classification has to decide whether an unvoiced dominant TFU originates from the singing voice or from another source. In the case of singing voice the TFU gets labeled voice and music otherwise.

In order to be able to discriminate between the classes a feature set is build. The author of [HJT08] proposed 36 MFCCs consisting of 12 MFCCs and their delta and double-delta values. To increase computational efficiency and decrease the necessary amount of data, we use the Magnitude Modulation Spectrum (MMS) [GOO06] instead. The MMS is based on the STFT interpretation as a subband filter-bank. Any subband output then corresponds to a time series describing the amplitude and phase evolution of a signal around the subband center frequency. The spectral analysis of this time series results in the modulation frequency domain. This domain reveals amplitude modulation components associated with the subband center frequency. As an example, if the input signal consists of a single sinusoid with constant amplitude and a frequency equal to the subband center frequency, the MMS would show only a DC component. If on the other hand, the sinusoid is modulated in amplitude, the MMS will show a peak at the modulation frequency. The Modulation Spectrum and its modification has been applied in speech processing. Speech signals exhibit most of their energy in the lower modulation frequency region around 3 to 4 Hz, which is the syllabic rate of spoken language. The energy above 16 Hz is typically minimal and unimportant for speech signals. The use of the MMS in case of unvoiced singing, comes from the assumption that unvoiced sounds in singing and speech are likely to be very similar. In contrast, unvoiced sounds, originating from instruments should have less energy in low modulation frequencies.

Therefore, the feature extraction consists of first lowpass-filtering and downsampling each gamma-tone filter output by factor of 16, to further decrease the amount of used information, then the frame-wise MMS are calculated as follows

$$X_{MMS}[l,b] = \left| \sum_{n=0}^{N-1} |b[n]| e^{-2\pi i l n/N} \right| \tag{31}$$

where $X_{MMS}[l,b]$ is the magnitude of modulation frequency l and n being the time index of the output of gamma-tone filter b. For each TFU 4 MMS coefficients are calculated, resulting in a modulation frequency range from 0-20Hz with a frequency resolution of 5Hz. Figure 21 shows 10 MMS for the second gamma-tone filter with a center frequency of $2.6kHz$ which is exited with the signals singing voice, accompaniment and speech.

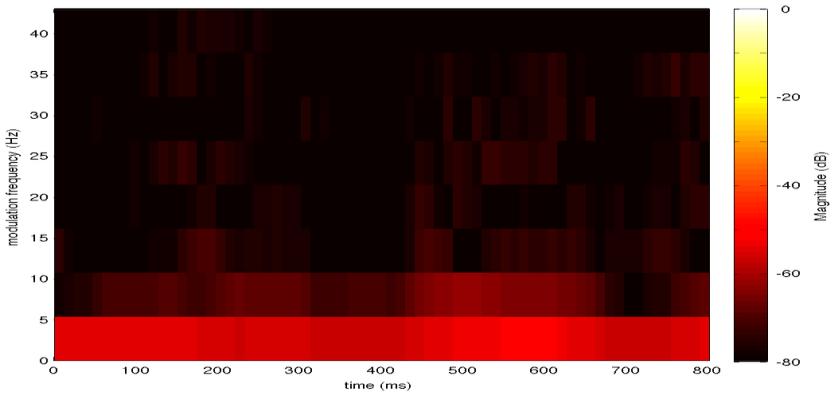
The similarity between speech and singing voice can be observed, as well as the lower energy for the accompaniment signal.

Next, similar to the voiced singing stage, a PCA is performed to transform the feature-set onto a orthogonal system. Finally, after the TFU classification using a SVM a binary mask is formed, which is post-processed. Due to the very short effective length of a TFU of $\sim$ 6ms the resulting mask is post-processed using the principles of Minimum Lifetime (MLT $= 20ms$) and Maximum Rest Time (MRT $= 20ms$). Figure 22 gives an example for postprocessing the classification results.

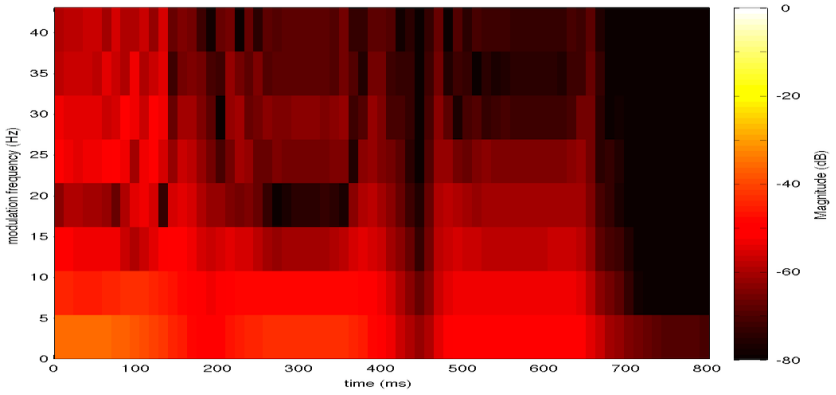
2.4.5 TFU Extraction

Once all TFU's are classified into the classes *voice* or *music*, those belonging to the class *voice* are extracted. To avoid disturbing noise at the beginning and ending of the extraction of a specific TFU, its amplitude is multiplied with a linear fade (duration $\sim$3ms). Since the TFU analysis is realized in overlapping frames with a hopsize of 256 samples ($\sim$6ms), each extracted TFU is of that length.

(a)



(b)



(c)

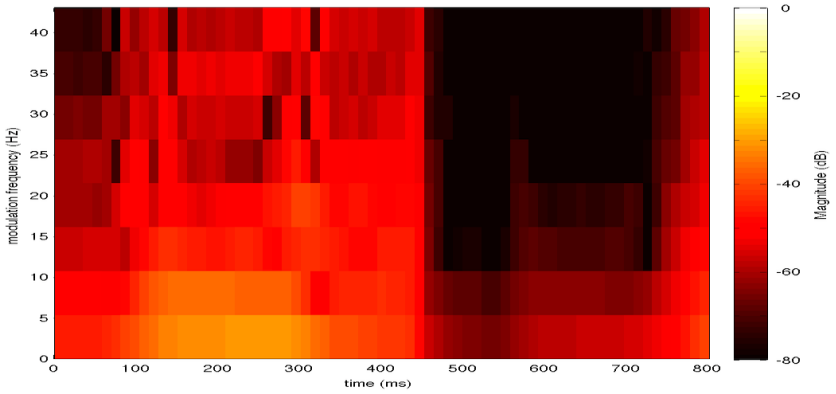
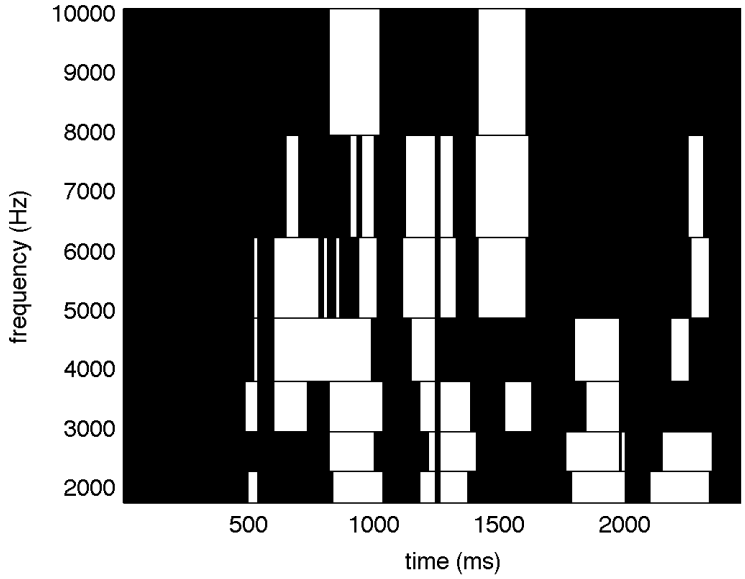


Figure 21: Example of Magnitude Modulation Spectrum for 3 different signals for gamma-tone filter No. 2 with center frequency $f_c = 2.6kHz$
(a) Accompaniment Signal
(b) Singing Voice Signal
(c) Speech Signal

(a)



(b)

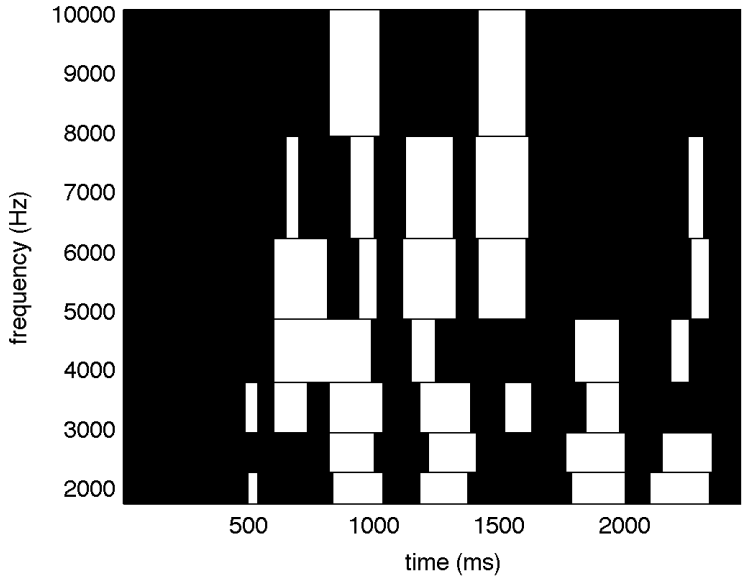


Figure 22: Postprocessing of Time-Frequency-Units (TFU) class labels, (white) TFU of class *voice*, (black) TFU of class *music*
(a)Classification Results
(b)After Postprocessing by Minimum Lifetime (MLT $= 20ms$) and Maximum Rest Time (MRT $= 20ms$)

3 Training of Classifiers

Due to the lack of datasets incorporating stereo audio-recordings with labeled vocal pitches, we had to create one. The training of the SVM classifiers for the voiced and unvoiced classification is based on this stereo-dataset. Additionally, to be able to present comparable results, a second dataset is used which consists of 9 mono audio recording from the MIREX 2005 training dataset - „vocal".

Both datasets will be described below:

Stereo Dataset

This data set consists of 6 stereo audio-recordings (16bit, 44.1kHz) in excerpts of 1-5 seconds incorporating the genres Pop, Rock and Jazz. All recordings include a stereo singing voice track as well as a stereo accompaniment track which are premixed with a Vocal to Accompaniment Ratio (VAR) of 0dB.

We define the VAR to be

$$VAR = 10log_{10}\frac{\sum |x_v[n]|^2}{\sum |x_a[n]|^2} \tag{32}$$

n being the discrete time index, $x_v[n]$ and $x_a[n]$ are, respectively, the time domain signals of singing voice and accompaniment. The VAR represents an average measure thus, depending on the considered time window the ratio will differ. If the VAR is calculated considering the full length time signals and set to be 0dB it is very likely that there will be shorter time fragments where the ratio reaches less than 0dB. The main reason being that the singing voice signal usually contains more silence than the accompaniment signal (figure 23).

Next, the vocal pitches are labeled in a two step process. The initial pitch labeling is performed in *Sonic Visualiser v1.9*[2] by applying an Audio Pitch Detector[3] which is based on the YIN Frequency Estimation method. The estimator settings are a window length of ∼46ms, a hopsize of 10ms and the vocal f_0 is restricted within the range 100Hz to 800Hz. Subsequently, the resulting frequency trajectory is manually corrected for octave errors.

2. Sonic Visualiser ©2005-2011 Chris Cannam and Queen Mary, University of London
3. Audio Pitch Detector v2, Marker: Paul Brossier (plugin by Chris Cannam)

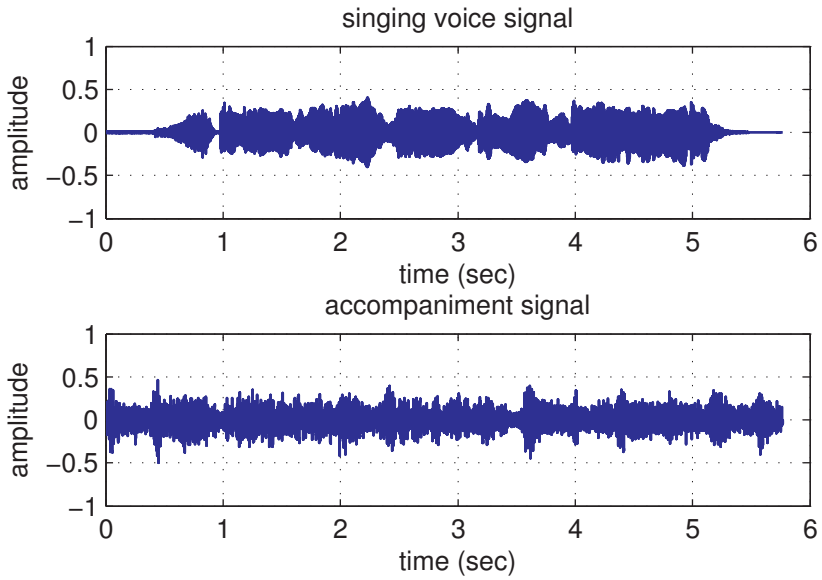


Figure 23: time domain signal of singing voice and accompaniment incorporating a Voice to Accompaniment Ratio (VAR) of 0dB

Mono Dataset

This data set consists of 9 Songs from the MIREX 2005 training data set „vocal“, which was originally used in the MIREX Audio Melody Extraction contest. All songs are mono audio-recordings (16bit, 44.1 kHz) in excerpts of 24-34 seconds in duration. This dataset include text-files with a manually annotated singing voice f_0 trajectory in time increments of 10ms, incorporating the genres Pop, Rock and Jazz.

SVM Parameters

The SVM classifiers for the Pitch Track classification and the Time-Frequency-Unit classification have to parametrized i.e, the Kernel function and its parameters has to be chosen. In general this can be done either automatically, i.e. by a search and optimizing algorithm, or manually. Since finding a suitable optimization algorithm can be very time consuming and therefor would exceed the timeframe for this thesis, we decided to follow the instructions given by Hsu, Chang and Lin [HCL03]. The authors propose to find the optimal parameter setting by following the subsequent steps:

1) Scaling of feature set
   Mainly, this is done to avoid attributes in smaller numeric ranges to be dominated by those in greater numeric ranges. Therefore, the feature set is linearly scaled to the range $[-1, +1]$. Note that this is necessary not only for the training set but

also the test set.

2) select Radial Basis Function (RBF) Kernel
   The authors argue this to be reasonable choice because the RBF kernel:
   - maps the feature values in a higher dimensional space, thus it enables good class separation even if the classes were initially not linear separable. Additionally the linear Kernel is a special case of the RBF kernel and the sigmoid kernel behaves like RBF for certain parameters.
   - has less hyperparameters as for example the polynomial kernel
   - has less numerical difficulties
3) Cross-Validation and Grid-Search

The following sections will describe the training procedure to train the classifiers for the voiced and the unvoiced classification stages.

3.1 Pitch Track Classification

Since the SVM classifies pitch tracks rather than individual pitches, the training data set (see page 41) incorporating the ground truth hast to be computed on basis of the reference files. Therefore, all songs are passed through the processing stages APP, MPE and PT. The resulting pitch tracks are then divided into two separate classes named *voice* and *music* based on the overlap with the reference f_0 trajectory. This concept was introduced by Rahimzadeh [Rah09] which we implemented in accordance. The decision boundaries for this assignment are shown in table 3.

overlap with reference	class assignment
$\geqslant$60%	voice
$\leqslant$10%	music
>10% and <60%	no assignment, tracks are removed from ground truth

Table 3: Class assignment of pitch tracks based on overlap with reference pitch trajectory

If the overlap is between 10 % and 60 % the class assignment is skipped and the corresponding pitch tracks are removed from the training data set. This is done to avoid unreliability of tracks. Additionally, a pitch track is considered overlapping with the reference, if its f_0 is within the range of $\pm$ 3% to the reference pitch trajectory. An example of such an assignment is shown in figure 24, along with the onset based pitch track splitting (see chapter 2.3.3, page 21) and its improvement in the class assignment task. As can be observed in figure 24(a) the first 2 pitch tracks overlapping with the reference pitch track are rejected. Now, using the onset based track splitting the tracks are divided into 4 separate tracks, where two tracks could be assigned to the class voice and one track to the class music.

The Features for both classes of the assigned pitch tracks are than extracted and stored in a database.

To make the features as robust and meaningful as possible the feature set proposed in section 2.3.4 is studied in detail to find a suitable subset. Many methods make use of the Fisher's Ratio (FR), which is a measure for the inter-class scatter in comparison to the intra-class scatter. The Fisher's Ratio for each feature fn can be calculated as follows

$$FR_{fn} = \frac{(\mu_{fn,music} - \mu_{fn,voice})^2}{\sigma^2_{fn,music} + \sigma^2_{fn,voice}} \tag{33}$$

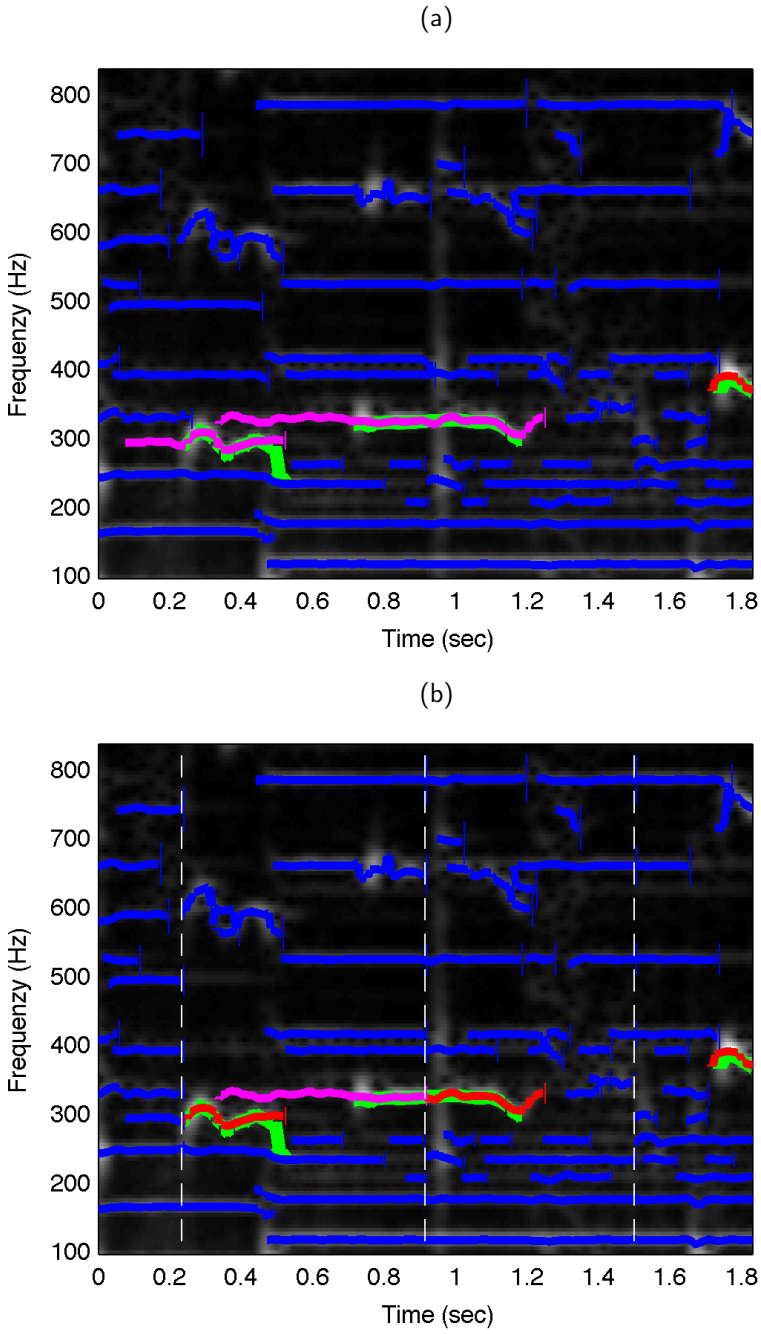


Figure 24: Class assignment of pitch tracks based on overlap with reference f_0 pitch track (green), class voice (red), class music (blue), rejected tracks (magenta), detected onsets (dashed line)
(a) class assignment without onset based pitch track splitting
(b) improvement in using onset based pitch track splitting

Figure 25 shows an example of the FR for 3 different feature value distributions. Note that for all examples the mean values remain constant, only the variances decrease. This results in a higher ratio, which is also reflected in the overlap of the two classes.

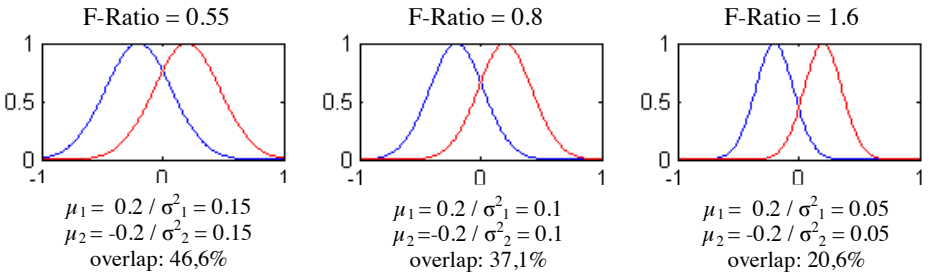


Figure 25: Fisher's Ratio for 3 different distributions and 2 different classes, Note that the Mean values remain constant while the variance decreases resulting in a higher Fisher's Ratio, from [Rah09]

The actual FR values for the proposed feature set are shown averaged over all songs of the stereo dataset in figure 26 and per song and feature in figure 27. The exact values are listed in Appendix A.

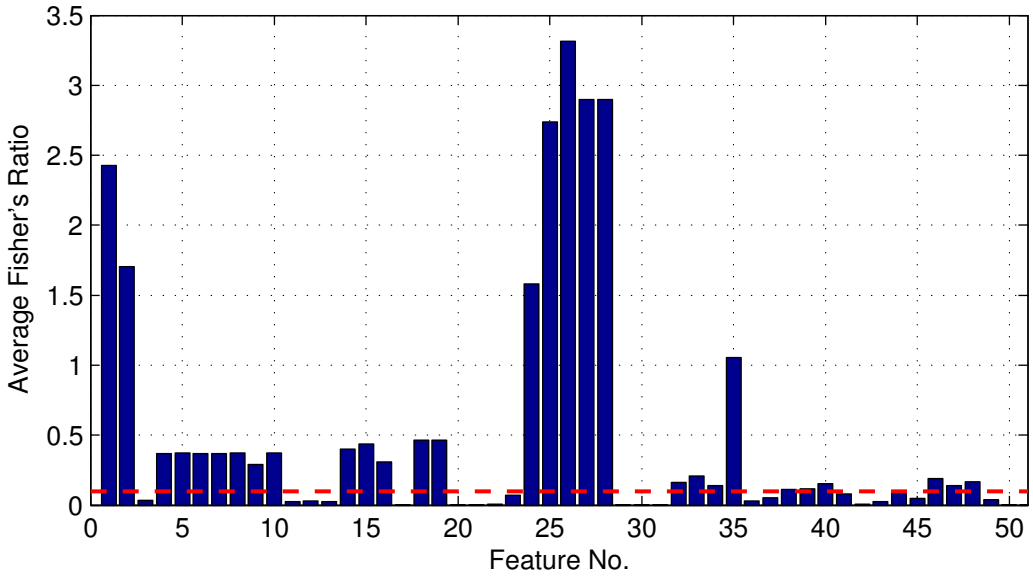


Figure 26: Average Fisher's Ratio for each feature of the stereo training dataset, the dashed red line represents the chosen threshold for excluding features

The threshold upon which a feature is excluded from the final sub set is chosen empirically to be 0.1. One might assume this to be a rather low value, but experiments showed that setting the threshold to higher FR values, decreased the classification performance, which might be unexpected at first. Theoretically, the higher the ratio, the more significance

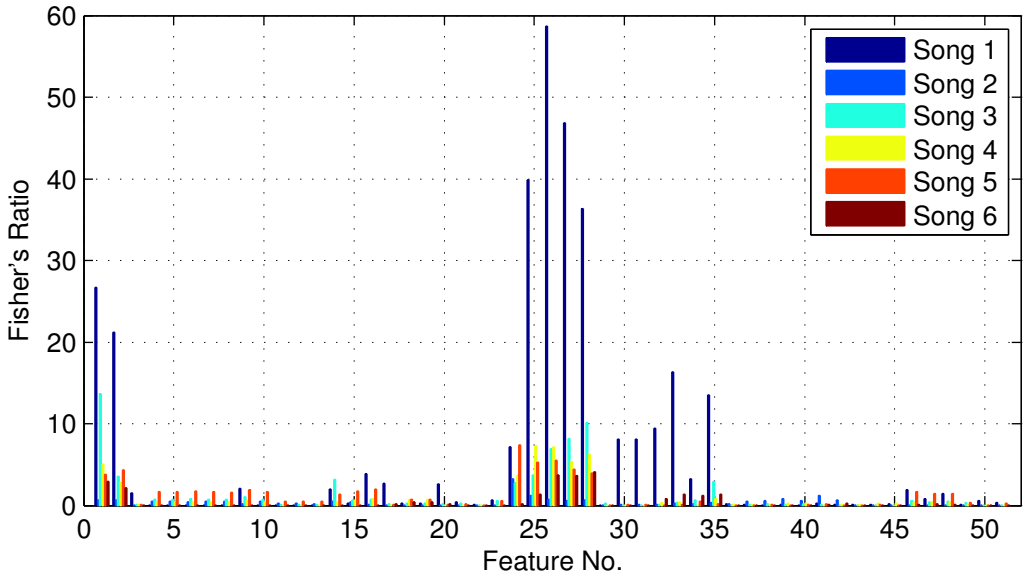


Figure 27: Fisher's Ratio for each feature and for each Song of the stereo training dataset

the resulting feature subset should have, i.e. the higher the discriminability. While this is generally true, tests showed that the information loss introduced by excluding more and more features prevails at a certain point, especially when using a powerful classifier like the SVM. Powerful refers to the SVM's capability to separate in non-linear feature spaces. The final feature subset is then obtained by performing a Principal Component Analysis (PCA) to transform the feature space onto an orthogonal system. It could be observed that the classification performance increases significantly (5-10%) by using PCA.

Next, the RBF parameters for the SVM have to be set. As mentioned on page 42, this is realized by Cross-Validation and Grid-Search. In accordance to the mentioned steps, first a coarse grid is used before performing a more precise grid search while incorporating a 6-fold cross-validation. After running the grid searches for each subset the mean values are computed and shown in figure 28.

The optimal parameters C and γ are chosen by maximizing

$$max(\text{total no\# of TP} - \text{total no\# of FP}) \tag{34}$$

where the total numbers of true positives TP and false positives FP are obtained by summation over all cross-validation subsets. The metrics along with the corresponding confusion matrix are presented on page 63.

Finally, the chosen parameters $C = 2^{14.75}$ and $\gamma = 2^{1}$ are used to pre-train the SVM,

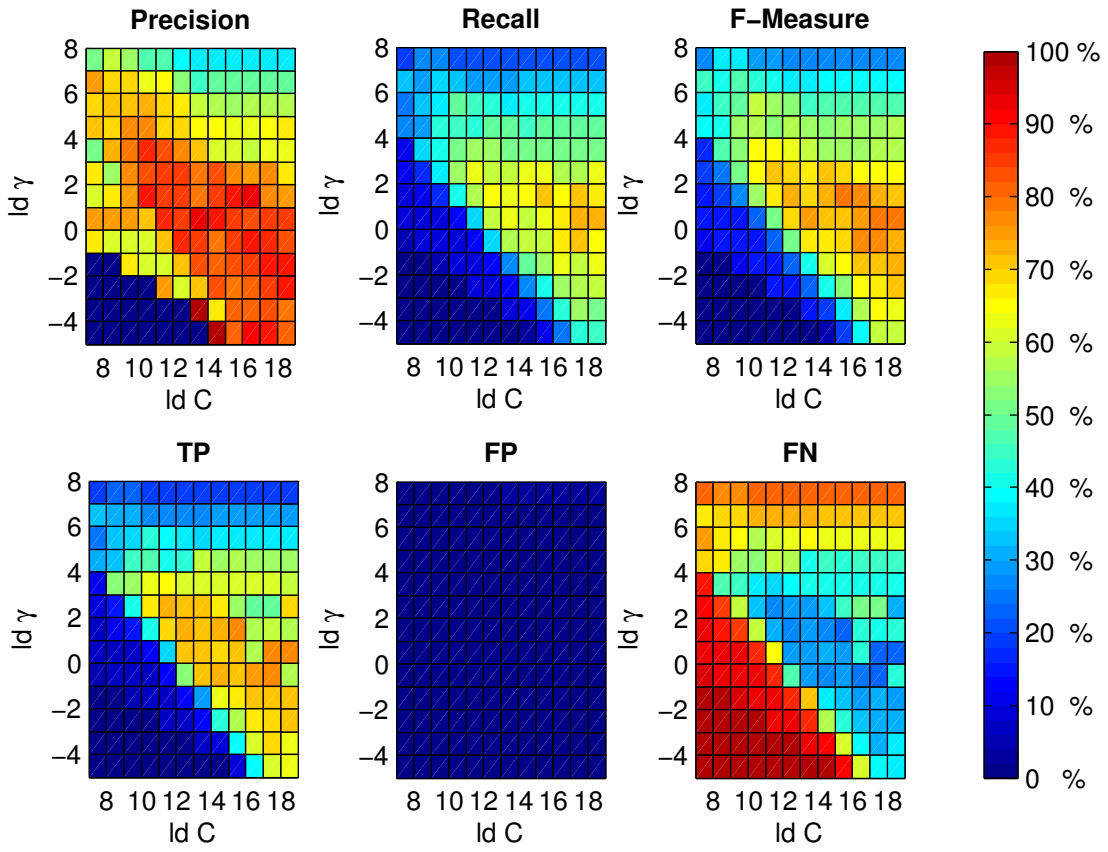


Figure 28: Grid Search Results for Pitch Track Classification, Average of 6-fold cross-validation

which decreases the amount of time necessary for the classification process.

3.2 Time-Frequency-Unit Classification

To our knowledge there is a lack of training-sets for unvoiced singing voice. Therefore, we used the previously presented stereo dataset (page 41). The necessary ground truth for the TFU classification is created in a three-fold process.

1) Preprocessing

The singing voice signals and accompaniment signals are separately available as stereo recordings. Based on a predefined Voice to Accompaniment Ratio (see eq. 32, page 41) of 0dB, both signals are mixed to build what we call the premix.

The singing voice signal is now processed by the stages panning index preprocessing and harmonic/percussive decomposition. Next, a moving average subtraction is performed on the percussive signal. The moving average subtraction (filter length 17 samples) can be seen as high-pass filtering with a filter cutoff frequency of $\sim$2.2 kHz. The filtering comes from the assumption, that unvoiced singing is not present in lower frequency regions. The filtered percussive signal then represents the unvoiced singing signal. Finally, the unvoiced singing is subtracted from the premix. The last mentioned signals, the unvoiced singing signal and the premix signal without the unvoiced singing, are used to build the ground truth for the TFU classification.

Finally, the above described method to obtain the unvoiced component of the singing voice signal is now applied to the premix signal. In this way the voiced and unvoiced components are separated. These will be used to build the ground truth for the unvoiced frame detection and referred to as unvoiced premix and voiced premix.

The next two paragraphs will describe how the specific labels are created based on energy considerations. The Energy E of a signal $x[n]$ is calculated as follows:

$$E(x[n]) = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2} \tag{35}$$

where N is the frame length in samples.

2) Unvoiced Dominant Frame Labels

The unvoiced premix and the voiced premix are compared frame-wise in energy to assign the label unvoiced dominant or voiced dominant. Additionally, the time instances a

singing voice f_0 is present, is taken into account. As mentioned, the time instances where a singing voice is detected by the voiced singing stages are extended by an additional time window (pre-/post-listen).

In summary, this leads to the following decision rule:

$$E(\text{premix}_{unvoiced}) > E(\text{premix}_{voiced}) \wedge \text{voice } f_0 \text{ present} \Rightarrow \text{unvoiced dominant frame} \tag{36}$$

3) TFU Class Labels

Now the energies of TFU's are compared. In particular, the TFU Energies resulting from unvoiced singing and the ones resulting from the premix without unvoiced singing. If the following rule

$$E(TFU, \text{unvoiced singing}) > E(TFU, \text{premix without unvoiced singing}) \tag{37}$$

holds true for a particular TFU, then it is labeled as class voice and as class music otherwise.

Processing the whole stereo dataset, 2278 training instances for the class *voice* and 2720 training instances for the class *music* are created.

SVM Training

Having build the ground truth, the SVM now is trained by:

1) panning index filtering and harmonic/percussive decomposition
2) moving average subtraction of percussive signal
3) splitting in frequency bands, using a gamma-tone filter-bank
4) time decomposition of every filter output in overlapping frames
5) extraction of 4 MMS coefficients (page 37) for each frame
6) perform PCA on the resulting feature set to transform it onto an orthogonal system
7) find the best SVM parameter set (page 42):
   - scaling of feature set
   - select RBF Kernel
   - Cross-Validation and Grid-Search (figure 29)

8) determination of the optimal RBF parameters, max(true positives - false positives), eq. 34 on page 47

The optimal SVM RBF parameters were found to be $C = 2^2$ and $\gamma = 2^5$.

It should be noted, that only one SVM model is trained for all TFU's. This means, the classifier does not distinguish between different gamma-tone filters.

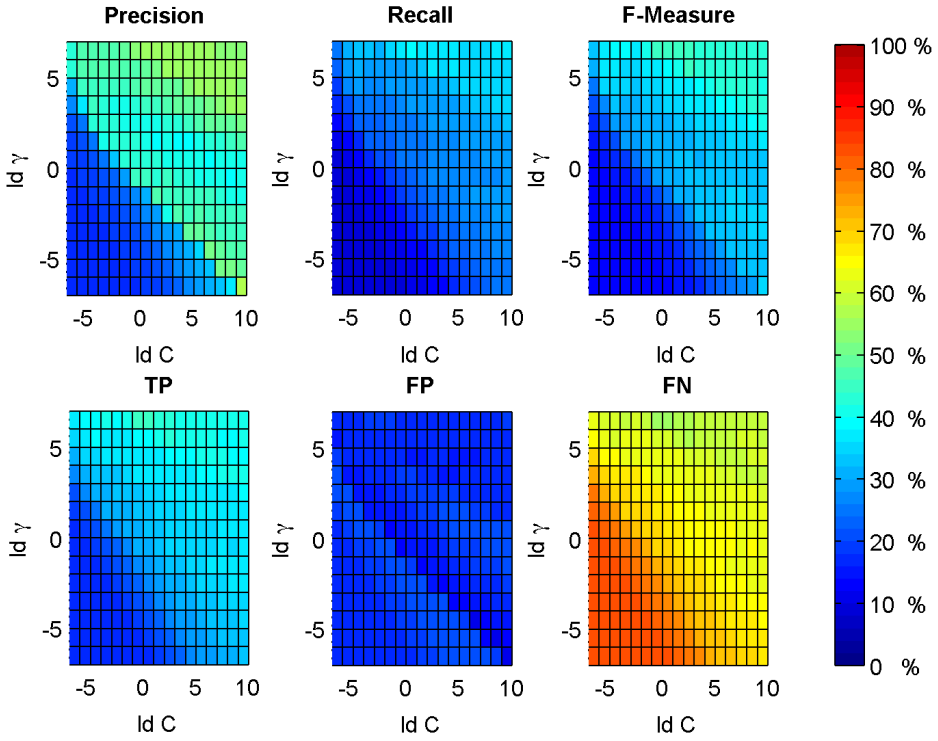


Figure 29: Grid Search Results for Time-Frequency-Unit Classification, Average of 6-fold Cross-Validation

4 Evaluation

Before we present a detailed evaluation, let us first revisit the evaluation measures namely Precision, Recall, F-Measure and Accuracy.

The Precision is defined by:

$$P = \frac{TP}{TP + FP} \tag{38}$$

The Recall is defined by:

$$R = \frac{TP}{TP + FN} \tag{39}$$

The F-measure is defined by:

$$F = 2\frac{P \cdot R}{P + R} \tag{40}$$

Finally, the Accuracy is defined by:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{41}$$

How the metrics TP, FP, TN and FN are obtained varies thus, will be explained for each evaluation section separately.

4.1 Panning Index Preprocessing

The panning index preprocessed signal $PI[n]$ and its effect on the subsequent processing stages is compared to its common alternative, usually referred to as Center Signal (CS). We define it to be

$$CS[n] = \frac{1}{2}\left(x_{left}[n] + x_{right}[n]\right) \tag{42}$$

representing a simple time domain summation of left channel x_{left} and right channel x_{right}.

As a quality measure, we study the Spectral Voice to Accompaniment Ratio (SVAR) resulting from the different processing techniques. We define the SVAR to be:

$$SVAR[k] = 10log_{10}\left(\frac{|FFT(x_v[n])|}{|FFT(x_a[n])|}\right) \tag{43}$$

where k is the frequency index. The singing voice signal and the accompaniment signal is passed separately through the panning index preprocessing resulting in $SVAR_{PI}$ and the center signal processing resulting in $SVAR_{CS}$.

First, let us consider a fixed window width Ψ_w for the PI preprocessing. The resulting SVAR's are shown in figure 30. Two observations can be made. First, both $SVAR$'s decrease to low frequencies. This comes from the fact, that instruments that are usually positioned in the center of the stereo-panorama, e.g. bass drum or double bass, introduce significant energy in low frequencies, while the singing voice usually has less energy in low frequencies (see figure 2). Second, the $SVAR_{PI}$ resulting from the panning index processing is higher than the one from the center signal and reaches a maximum value of 8dB. This maximum value however is dependent on the chosen window width. To illustrate this dependency the $\Delta SVAR$, i.e. the difference in SVAR resulting from panning index preprocessing and center signal processing, is shown in figure 31.

The $\Delta SVAR$ is calculated as follows:

$$\Delta SVAR = SVAR_{PI(\Psi_w)} - SVAR_{CS} \tag{44}$$

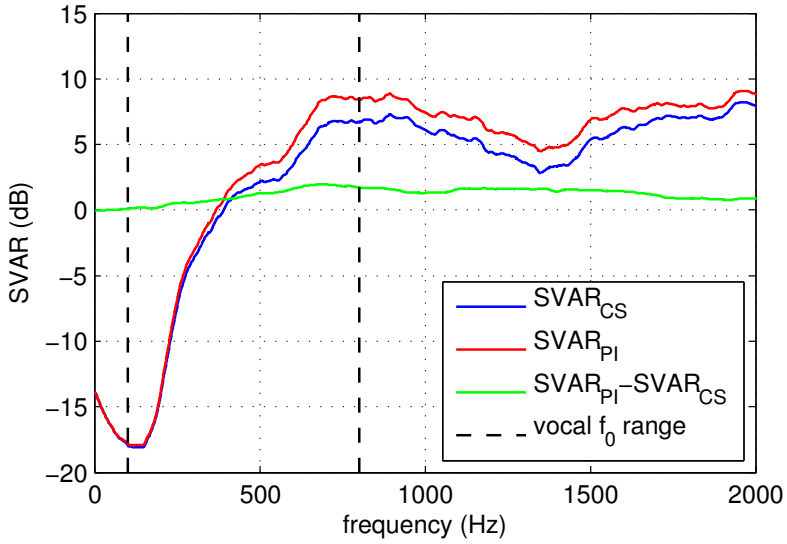


Figure 30: Spectral Voice to Accompaniment Ratio for Center Signal and Panning Index Preprocessing, Ψ_w =0.6

It can be seen that, as the window width decreases the $\Delta SVAR$ increases, while if the window width increases the $\Delta SVAR$ decreases. This behavior is expected, since the wider the window, the more energy from the accompaniment signal is preserved. However, it could be observed that by choosing a narrower window, not only the $SVAR_{PI}$ increases, but the singing voice signal gets affected as well. If the singing voice is perfectly positioned in the center and moreover does not vary in position over time, it is not affected by the chosen window width. However, if audio effects are applied to the

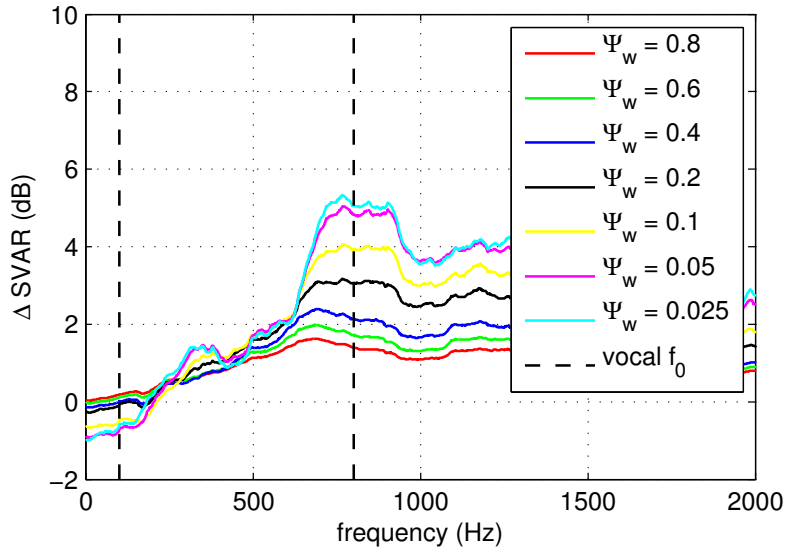


Figure 31: Δ Spectral Voice to Accompaniment Ratio (SVAR), calculated as the difference in SVAR between panning preprocessing and center signal processing

singing voice signal, e.g. reverb or delay based effects, its position in the panorama changes slightly over time. Hence, the narrower the window the more likely the singing voice is affected by the panning preprocessing by a significant amount. This effect is quantified by comparing the panning preprocessing to the center signal processing where both techniques are applied solely on the singing voice signal. Firstly a spectral comparison is realized by using the Panning Index to Center Signal Ratio (similar to eq. 43):

$$PICSR[k] = 10log_{10}\left(\frac{|FFT(PI_v[n])|}{|FFT(CS_v[n])|}\right) \tag{45}$$

where the subscript v marks the singing voice as basis for the different processing techniques. Figure 32 shows that the PICSR decreases for narrower windows. The fact that for Ψ_w >0.2 the PICSR exceeds 0dB originates in the center signal processing, due to the time domain summation of out of phase components and the resulting loss of energy. To stress the fact that a narrow window evokes not just frequency dependent damping of the singing voice signal, figure 33 depicts the loss of correlation by choosing a small Ψ_w. The degree of correlation is represented by the correlation coefficient, which is calculated by:

$$CCoeff = \frac{\sum^{n}(PI_v[n] - \overline{PI_v})(CS_v[n] - \overline{CS_v})}{\sqrt{\sum(PI_v[n] - \overline{PI_v})^2 \sum(CS_v[n] - \overline{CS_v})^2}} \tag{46}$$

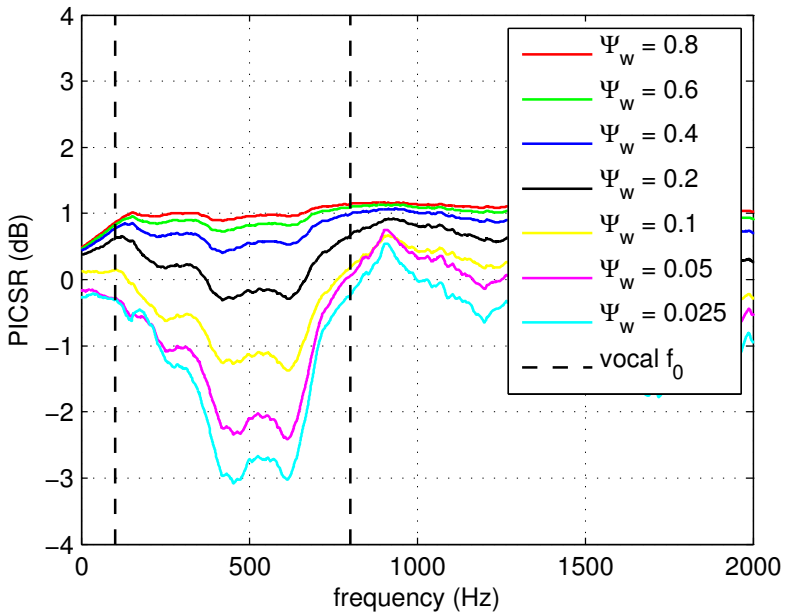


Figure 32: Panning Index to Center Signal Ratio (PICSR) and its dependency on the panning index window width, PICSR calculation is based solely on singing voice signals

The presented evaluation results conclude that, the optimal window width should be as narrow as possible to maximize the improvement in using the panning index preprocessing, i.e. maximizing $\Delta SVAR$. At the same time, the window width should be as wide as necessary to minimize information loss induced by the processing. As can be expected Ψ_w heavily effects the subsequent singing processing, thus the final decision on the exact value is presented in the pitch track classification evaluation, chapter 4.2.3 on page 64.

It should be mentioned that the selected VAR, used in premixing singing voice and accompaniment, does not have any effect on the resulting $\Delta SVAR$ values.

4.2 Voiced Singing Voice

The evaluation of the voiced singing stages is based on the stereo data set presented on page 41, which was created for the purpose of this thesis. Additionally, 9 songs from the MIREX 2005 training set (page 44) are used solely to compare against the results achieved by A. Rahimzadeh.

4.2.1 Multi Pitch Estimation

To evaluate the performance, the Raw f_0 Accuracy is used which can be calculated as follows:

$$RawAcc_{f_0} = \frac{\text{no\# of correctly estimated vocal } f_0\text{'s}}{\text{total no\# of frames containing singing voice}}(\%) \tag{47}$$

A pitch is considered estimated correctly if its frequency is within $\pm\frac{1}{4}$ -note ($\approx$ 3%) to the reference f_0.

The accuracy of the estimation process is mainly dependent on the number of extracted estimates per frame, due to the fact, that the estimation relies on the vocal pitch to be prominent in the magnitude spectrum. The relationship between the amount of pitch estimates and the Raw Estimation Accuracy is shown in figure 34. As expected, the more estimates being extracted, the higher the estimation accuracy which reaches its maximum median value of 92.6% (CI $\pm$3.5%) for 15 and more estimates. This leaves $\sim$ 7.4% of the vocal pitches undetected which could have 2 main reasons. Firstly, spectral leakage and the resulting concealment of peaks in the magnitude spectrum and second, some of the vocal pitches occur with very low amplitudes. Latter makes it very difficult to detect the vocal pitches, especially if other sources have significant energy in the same frequency region. Furthermore, figure 34 shows that if the amount of pitch estimates is restricted to very few estimates, the estimation accuracy decreases rapidly while its variance increases, since it is very likely that strong spectral components coming from other sources will be detected instead.

In order to choose the optimal number of estimates one has to consider not only its effect on the pitch estimation accuracy but on the subsequent processing. As already mentioned, the ground truth used to train the pitch track classifier has to be computed on the basis of the known reference vocal pitches. Therefore, the pitch estimation not only effects the pitch track classification, but its training, i.e. the ground truth. Thus, a way to maximize the ground truth is to maximize the significance of the resulting pitch

trajectories used in training. By significance we mean the amount of valid vocal pitches a pitch track, assigned to the class voice, contains. Maximizing the significance of the pitch trajectories leads to maximizing the meaningfulness of the resulting feature values and thereby allowing optimal classification performance. Thus, a measure is needed to decide on the number pitch estimates, which we call the Raw Track Accuracy $RawAcc_{track}$.

It is defined as the number of correctly estimated vocal pitches divided by the true amount of vocal pitches.

$$RawAcc_{track} = \frac{\text{no\# of valid pitches in voiced ground truth}}{\text{true no\# of frames with active singing voice}}(\%) \tag{48}$$

The goal is to have pitch tracks containing a maximum number of true vocal pitches. The relationship between the track accuracy and the number of estimates is presented in figure 35. First of all, it is important to note that the Raw Track Accuracy only increases the reliability of the vocal pitch tracks, while having no effect on the pitch tracks of class *music*. Therefore, it gives more a qualitative statement. Nevertheless, the large variances for few estimates as well as for many illustrate that the number of estimates does not affect all tracks. For the case of very few estimates, figure 35 illustrates that for some songs there are other musical sources with significant energy in the vocal f_0 frequency range, thus the sources with higher amplitudes will be extracted instead of the vocal pitches. The large variances for many extracted estimates show that there are sources with a high amount of energy in close proximity to the vocal f_0 trajectory. A good trade off seems 10 estimates, since the variance seems acceptable and at the same time the median accuracy is still high.

Initially more parameters were implemented to tune the MPE performance, e.g. the number of partials that have to be detected, the allowed number of undetected partials and the allowed deviation of partial frequencies to their perfect harmonic location. The motivation was to pre-filter or decrease the number of estimates while preserving the maximum amount of valid vocal pitch estimates. However, these concepts failed since the identification of a partial, i.e. identifying the corresponding f_0, and proved to be unreliable. The first step would be, to have reliable information about the existence of a particular partial, which again could not be achieved. To elaborate, let us consider the partial frequency deviation. A partial frequency deviation of 0% means, that the occurring magnitude at each multiple of f_0 is considered to be a partial, regardless if a local maximum could be detected. In contrast, if we allow a deviation, a local maximum hast to be detected, otherwise the corresponding partial is considered to be missing. Ideally we would always detect a partial if present, but practically this assumption holds

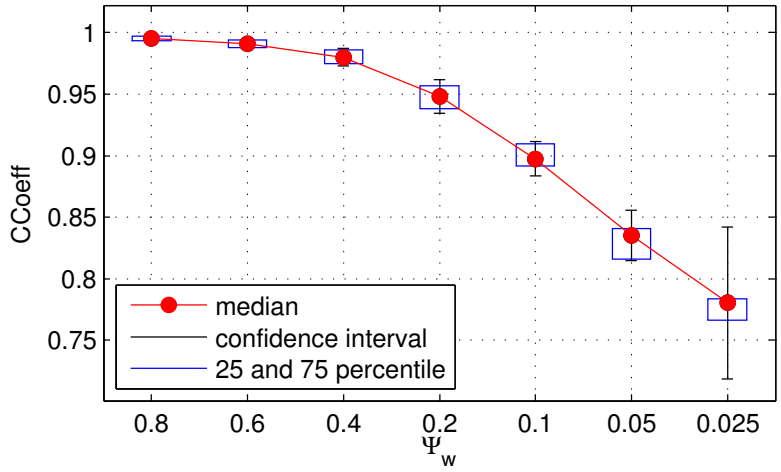


Figure 33: Correlation coefficient (CCoeff) between panning preprocessed singing voice and center signal processed singing voice

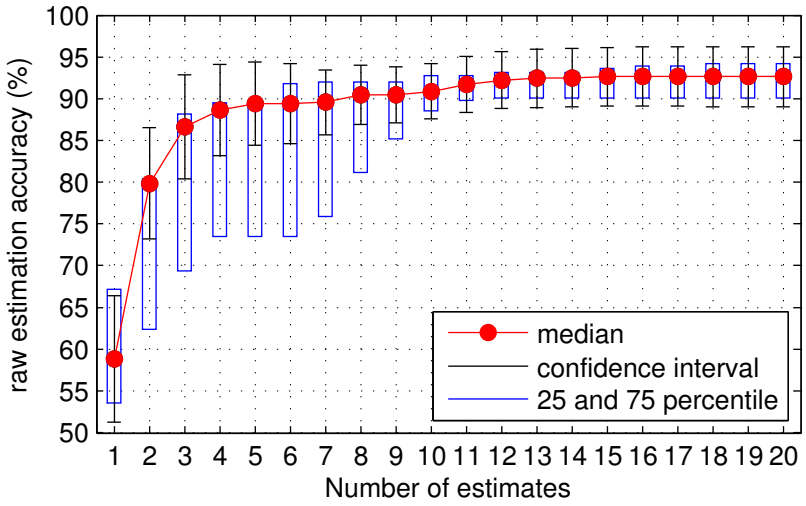


Figure 34: Number of extracted pitch estimates per frame versus Raw f_0 Estimation Accuracy

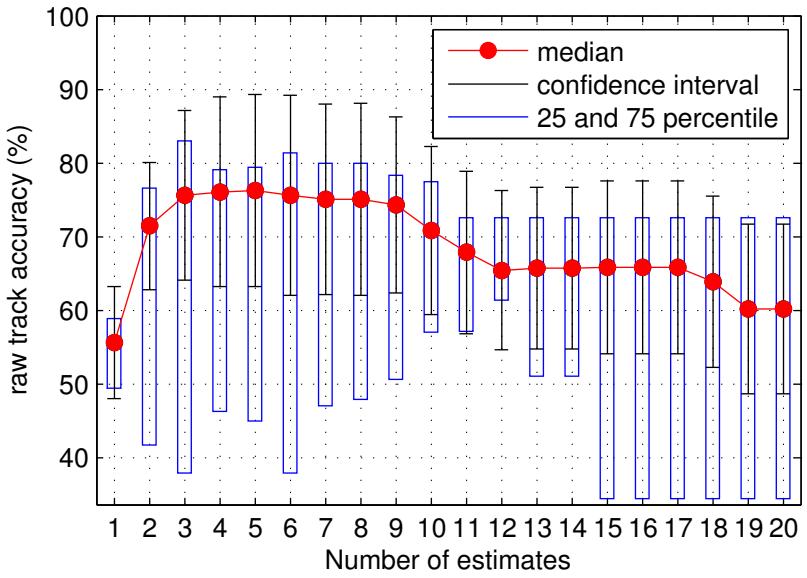


Figure 35: Number of pitch estimates per frame versus Raw Track Accuracy

not true, since we are restricted at least by the number of present sources in a mix and their (partially) overlapping spectra. Experimental results confirmed the unreliability in detecting the presence of a partial thus, the restriction in the amount of detected partials along with their frequency location is left out of the process. Yet, we believe that having a reliable mapping of partials to their corresponding f_0 would improve the pitch track classification performance significantly. Since the frequency location of partials, their magnitude and evolution over time is one important distinctive property inherent to a class of sources.

To conclude the evaluation, we present in table 4 the specific Raw Estimation Accuracies in comparison to Rahimzadeh. Apart from minor individual variations the same overall estimation accuracy is achieved.

file name	proposed	Rahimzadeh	Δ
train01	97.7	96.9	0.8
train02	85.8	87.9	-2.1
train03	80.3	81.1	-0.8
train04	89.2	88.6	0.6
train05	98.0	97.1	0.9
train06	81.0	83.7	-2.7
train07	93.5	92.1	1.4
train08	95.6	93.1	2.5
train09	92.9	92.9	0
mean	90.45	90.38	0.06

Table 4: Multi Pitch Estimation Raw Accuracy in comparison to A. Rahimzadeh, all values in %, evaluated on 9 songs MIREX 2005 training data set „vocal“ (see page 42)

4.2.2 Pitch and Partial Tracking

To evaluate the performance of the pitch tracker and illustrate how much information is retained, i.e. the amount of correct vocal pitches, we present the Relative Pitch Tracking Accuracy:

$$RelPitchAcc = \frac{\text{no\# correct vocal} f_0\text{'s in all pitch tracks}}{\text{total no\# of correctly estimated vocal} f_0\text{'s}}(\%) \tag{49}$$

Again, like in the MPE evaluation, a pitch is considered correctly estimated if its frequency is within $\pm\frac{1}{4}$ -note ($\approx$ 3%) to the reference f_0. The Name Relative Accuracy signifies the fact that the amount of correct vocal f_0's contained by all pitch tracks is measured in relationship to the correctly estimated f_0's coming from the MPE. In that way a Relative Pitch Tracking Accuracy of 100% means that none of the correct vocal f_0's was discarded in the pitch tracking process.

Apart from the post-processing of pitch tracks, i.e. discarding unreliable pitch tracks because of low MLT or low f_0 salience (see page 21), the achievable accuracy mainly depends on the chosen frequency deviation Δ_f which a pitch track is allowed to change from frame to frame. As a reminder, Δ_f additionally defines the frequency range surrounding the pitch track f_0, within pitch estimates are considered to be possible candidates for the particular pitch track.

Figure 36 shows the relationship between Δ_f and the RelPitchAcc. A maximum median value of 97% is achieved for Δ_f =4%, thus only 3% of the correct pitches are discarded. Although 3% of the correct vocal pitches are lost, at the same time the post-processing discards an average value of 73% (MLT=68%, f_0 salience=5%) off all

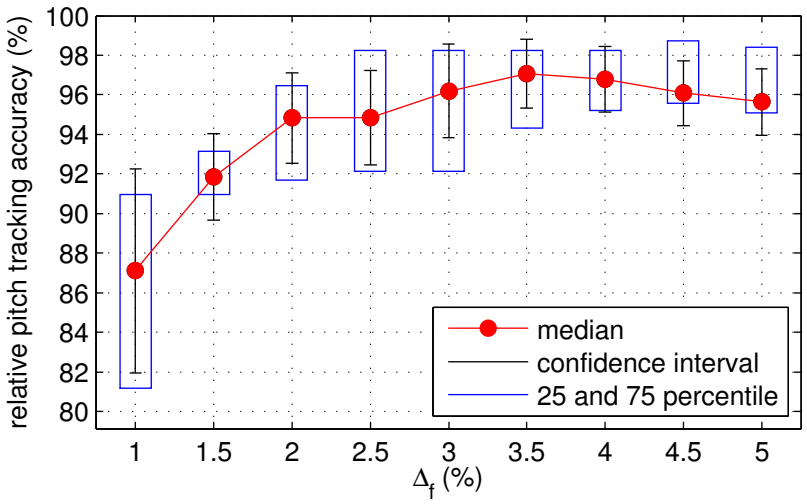


Figure 36: Relative Pitch Tracking Accuracy versus frequency deviation Δ_f

pitch tracks increasing significantly the meaningfulness of the resulting feature set. The decreasing RelPitchAcc for small Δ_f values originates in the fact, that the pitch tracks are terminated more quickly, if the frequency range within possible pitch candidates have to be located is heavily restricted. Thus, the lower the Δ_f the shorter the pitch tracks become and the more pitch tracks will be discarded because they do not exceed the Minimum Life Time. In contrast, for high Δ_f's the mentioned frequency range widens and the more likely that a pitch candidate coming from another source are assigned to the vocal pitch track, especially if this source has significant energy in the vocal f_0 frequency range. The implemented Δ_f value was consequently set to 4%.

To conclude, we present a comparison based on the MIREX 2005 training set, of the proposed pitch tracker to the one proposed by Rahimzadeh. Rahimzadeh introduced a pitch tracker containing linear prediction, which intended to increase the performance, i.e. increase the pitch track reliability, such that the pitch nearest to the predicted pitch track frequency is assigned to the track. First, we changed the application of the LPC such that, the frequency range within we consider possible pitch candidates is extended to include the predicted track frequency. Experimental results showed, that the prediction can not be considered reliable and furthermore decreased the raw pitch tracking accuracy. The comparison between the accuracy achieved by Rahimzadeh, the studied modified LPC and the final proposed pitch tracker without prediction is shown in figure 37. The Raw Pitch Tracking Accuracy is similar to eq. 49 except correct vocal

f_0's are now set in relationship to the total nunmber of frames containing singing voice:

$$RawPitchAcc = \frac{\text{no\# correct vocal } f_0\text{'s in all pitch tracks}}{\text{total no\# of frames containing singing voice}}(\%) \tag{50}$$

Figure 37 shows, that the proposed pitch tracker outperforms Rahimzadeh's pitch tracker on all of the 9 songs from the MIREX 2005 training set by a mean value of ~7%. The biggest improvement is for song 4 with 12%, while the lowest is for song 9 with 3%. The mean RawPitchAcc of the proposed pitch tracker reaches 90%.

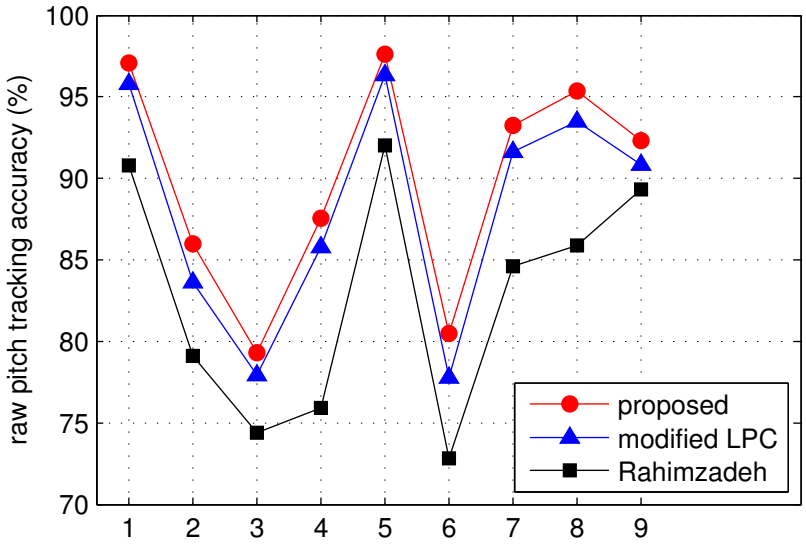


Figure 37: pitch tracking accuracy in comparison to A.Rahimzadeh, evaluated on 9 songs from the MIREX 2005 training data set (see page 42)

4.2.3 Pitch Track Classification

The Pitch Track Classification is evaluated in 2 steps. First, the benefit in using Panning Index Preprocessing is shown and second, the achieved accuracies are compared to Rahimzadeh.

As a measure for the classification performance, we define the Raw Classification Accuracy using eq. 41. The term „Raw“ states, that the metrics are calculated pitch wise based on the known reference f_0 while evaluating all classified pitch tracks.

Additionally, to focus on the singing voice pitch track we define the Final f_0 Raw Accuracy as the sum of TP and TN divided by the total amount of frames:

$$Final f_0 RawAcc = \frac{TP + TN}{\text{total no\# frames}} \tag{51}$$

where „Final“ states the fact, that it is calculated solely based on the final singing voice pitch track after post-processing the classification results.

The confusion matrix presented in table 5 shows the assignment of the metrics. The specific values can be calculated as follows:

$$TP = \frac{\text{no\# of correctly classified vocal pitches}}{\text{total no\# frames containing singing voice}} \tag{52}$$

$$FP = \frac{\text{no\# of pitches misclassified as vocal pitches}}{\text{total no\# frames not containing singing voice}} \tag{53}$$

$$TN = \frac{\text{no\# of correctly classified musical pitches}}{\text{total no\# frames not containing singing voice}} \tag{54}$$

$$FN = \frac{\text{no\# of pitches misclassified as musical pitches}}{\text{total no\# frames containing singing voice}} \tag{55}$$

	predicted class	
	voice	music
ground truth		
voice	TP	FN
music	FP	TN

Table 5: Confusion matrix for pitch track classification, voiced singing

Table 5 shows, that the assignment of the metrics is based on the underlying ground

truth. In this way, TP and FN as well as TN and FP will sum up to 100%.

Influence of Panning Index Preprocessing

To evaluate the effect of the panning index preprocessing on the classification, the classifier is trained on the stereo dataset presented in chapter 3 on page 41 and evaluated with 6-fold cross-validation. This data set incorporates 36 training instances for the class voice and 692 instances for the class music.

Table 6 presents the Δ Mean Accuracies between panning preprocessing and center signal processing, which are generally calculated by:

$$\Delta Acc = Acc_{PI} - Acc_{CS} \tag{56}$$

	ΔAcc					
	MPE	**PT**	**Raw Class. Results**			**Final f_0**
Ψ_w	**Raw Acc.**	**Raw Acc.**	**P**	**R**	**F**	**Raw Acc.**
0.1	-3.50	-5.92	-8.33	14.20	7.60	3.94
0.2	-1.42	-2.40	-1.35	-1.02	3.05	0.36
0.3	-0.16	-1.35	-6.22	11.51	8.25	6.93
0.4	-0.26	-0.94	-1.53	12.70	10.93	9.27
0.5	**-0.13**	**-0.86**	**1.33**	**15.30**	**14.17**	**12.13**
0.6	-0.12	-0.95	1.07	14.80	13.22	11.24
0.7	-0.02	-0.85	-4.23	14.98	9.77	9.57
0.8	0.37	0.11	-2.75	16.53	9.83	7.81

Table 6: Δ Mean Accuracies between panning index preprocessing and center signal processing depending on window width Ψ_w, all values in %, 6-fold cross-validation used

Leaving the narrowest window size ($\Psi_w = 0.1$) aside, one can observe the Final f_0 Raw Accuracy increasing for wider windows, reaching its maximum mean value of $\sim$12% for $\Psi_w = 0.5$ and then decreasing to even wider windows. As already mentioned in chapter 2.2, very narrow windows effect the singing voice signal if its position in the stereo-panorama varies slightly over time. The MPE and PT Accuracy for ($\Psi_w \leqslant 0.2$) reflect this fact. On the other hand, for very wide windows, more of the accompaniment signal is preserved thus making the classification process harder. For $0.3 \leqslant \Psi_w \leqslant 0.4$ the accompaniment signal is suppressed strongly, leading to rather short pitch tracks for this class and thereby resulting in features values which are difficult to distinguish from

the ones of class *voice*. A similar effect can be observed for very wide windows. Here the pitch tracks of class *music* are longer, but the increasing amount of preserved energy decreases the significance of the feature set. The underlying two main properties, the feature set is aimed to capture, are the f_0 variability and the relationship between partials and their corresponding fundamental frequencies. Since the partial frequency locations are restricted to be strictly harmonic, i.e. integer multiples of f_0, the more energy is preserved from the accompaniment the more likely that the partials and their extracted features are getting unreliable. The lower accuracy for narrow and wide windows is also reflected in fact, that more features are excluded from the final subset because they examine a low Fisher's Ratio. Another interesting finding is, that the pitch estimation and tracking accuracy does not change significantly while the Final Raw Accuracy does. This emphasizes the fact, that a high estimation accuracy does not guarantee high classification accuracy.

	MPE	PT	Raw Class. Results			Final F_0 raw acc.		
Song ID	raw acc.	raw acc.	P	R	F	PI	CS	PI-CS
1	74.1	71.8	44.1	49.2	46.5	55.3	56.7	**-1.4**
2	93.9	89.3	74.4	72	73.2	64.3	62.1	**2.2**
3	93.1	93.1	80.6	85.9	83.2	84.7	78.5	**6.2**
4	92	90.4	78.7	83.8	81.2	74.4	56.5	**17.9**
5	90.3	87.4	76.6	92.6	83.9	73.5	54.8	**18.7**
6	88.4	82.3	80.2	63.3	70.7	64.1	35	**29.1**
median	91.2	88.4	77.7	77.9	77.2	68.9	56.6	**12.3**
mean	88.6	85.7	72.4	74.5	73.1	69.4	57.3	**12.1**
CI	$\pm$6.5	$\pm$6.8	$\pm$12.4	$\pm$14.2	$\pm$12.4	$\pm$9	$\pm$12.3	**-3.3**

Table 7: Accuracy evaluations results for panning index preprocessing window width Ψ_w =0.5, all values in %, 6-fold cross-validation used on stereo dataset described on page 52

Now that we can set the window width to $\Psi_w = 0.5$, let us view the detailed performance for this particular setting, which is shown in table 7. The highest improvement is obviously achieved for the last 3 songs of the data set with up to $\sim$29%. The main reason why the panning index preprocessing allows better classification accuracy for these songs, lies in the fact that the panning index preprocessing is able to suppress the concurring sources and more importantly their partials by a higher amount than the center signal processing. As already mentioned, the relationship of partials to their fundamental frequency is an important property captured by the feature set. Thus, damping the

amplitudes of partials increases the discriminability between the two pitch track classes significantly.

The mean final final f_0 accuracy for the presented data set reaches ∼70%, which might be considered a medium performance. However, it is important to point out two significant facts. First, this data set only contains 17 seconds of audio in total, resulting in ∼730 training instances where only 36 are of class voice. This represents a very small data set, thus one can assume higher performance by incorporating more training instances. Second, the data set was created by premixing the singing voice and accompaniment signals with a VAR of 0dB. This is a rather unusual ratio (normally +2 to +5dB) which increases the difficulty for the classification process.

Classification Accuracy comparison

Table 8 shows the Raw Classification Accuracy achieved by Rahimzadeh compared to the proposed method. A significant improvement in accuracy is achieved with a maximum increase of 50%. The mean accuracy is improved by 16.3%, at the same time the confidence interval (CI) is reduced by 7%. It should be mentioned that Rahimzadeh used a k-nearest neighbor classifier which represents a very simple classifier compared to the SVM.

Song	Raw Classification Accuracy		
ID	Proposed	Rahimzadeh	Proposed-Rahimzadeh
1	95.6	89.8	5.8
2	84.9	66.3	18.6
3	75.6	55.1	20.5
4	82.1	69.2	12.9
5	91.3	88.4	2.9
6	90.4	40.2	50.2
7	93.1	78.9	14.2
8	91.5	83.6	7.9
9	86.0	66.5	19.5
median	90.4	69.2	**23.9**
mean	87.8	71.5	**16.3**
CI	±4.3	±11.3	**-7.0**

Table 8: Raw Classification Accuracy comparison of the proposed method to Rahimzadeh, all values in %, 9-fold cross-validation on „MIREX 2005 training data set - vocal“

4.2.4 Re-Synthesis

Due to the lack of a suitable data set to evaluate the performance of this stage, we present the examined methods to increase the audible re-synthesis quality.

The main goal was to find a property, that allows to identify partials and their corresponding fundamental frequency. The Motivation behind it was to exclude or discard partials if their frequency or amplitude is uncorrelated with the corresponding f_0. Therefore, correlation based properties were studied. The occurring problem was to detect and identify a set of corresponding partials in a reliable manner, only then particular properties can be studied. Experiments showed that the presented pitch estimation is not able to accomplished this task, mainly because of concurring sources and their resulting overlapping spectra.

Additionally, another method was studied, presented at the DAFX 2011 by Estefania Cano et. al. [CSD10]. They proposed the use of the phase information in source separation applications. The idea is, that partials from the same source not only relate in frequency (frequency-locking), e.g. partial frequencies are at integer multiples of f_0, but in phase as well (phase-locking). They showed the informative character of this so called phase-locking property for solo recordings of trumpet, clarinet, violin and piano. We studied the phase-locking for the case of singing voice. Experiments with solo singing voice recordings showed, that if a single note is sung, the phase values of partials exhibit similar behavior. However, if vibrato is introduced the phase-locking property seems no longer to be true, at least not in a strict sense. Therefore, further work has to be done in order to integrate this property into the presented framework.

As a consequence of the experimental results, all methods presented here were rejected because of their unreliability.

4.3 Unvoiced Singing Voice

4.3.1 Unvoiced Dominant Frame Detection

The UVFD accuracy is presented here in dependency on the LP error threshold. The question is, how well the LP error criteria, in conjunction with the presence of a singing voice f_0, is able to map the measured frame energies for unvoiced dominant frames, i.e. the ability to map the ground truth. The Accuracy is calculated using eq. 41 along with the following confusion matrix:

	assigned class	
	unvoiced	voiced
ground truth		
unvoiced	TP	FN
voiced	FP	TN

Table 9: Confusion matrix for Unvoiced Dominant Frame Detection

As figure 38 shows, the median accuracy reaches its maximum value of 83% (max 86.6%, min 43%) for a LP error threshold of $10^{-4.25}$. For this threshold all songs achieve a minimum accuracy of 72%, except for song 2 (refer to table 10). For this particular song the LP error seems to be a insufficient description. The reason being, that the LP error variances are very low, especially in contrast to the other songs in the set. Thus, the overall LP error threshold is not reflecting these particular low values, which leads to a low UVFD accuracy.

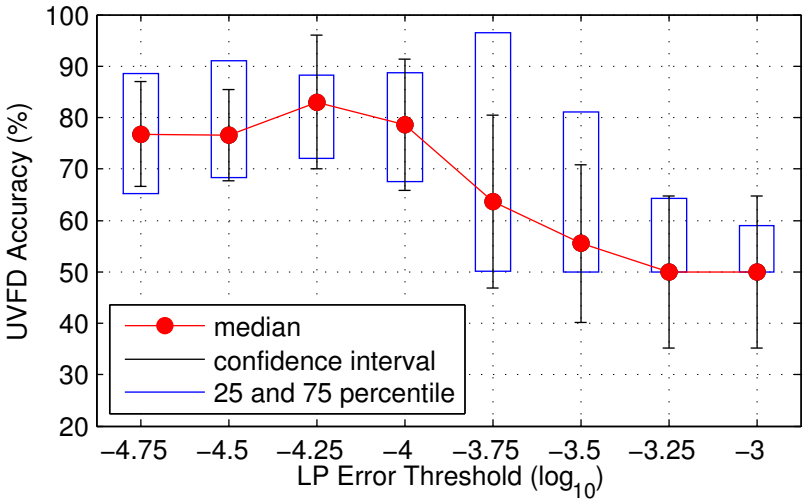


Figure 38: Unvoiced Dominant Frame Detection Accuracy versus threshold on Linear Prediction error

Next, the Panning Index Preprocessing and its influence on the UFVD Accuracy is examined (table 10). To allow a fair comparison, the accuracy results achieved by Center Signal Preprocessing are based on the ground truth, which was rebuild using the center signal. The overall median accuracy does not change significantly if Panning Index Preprocessing is performed. However, for song 6 and the case of center signal preprocessing the unvoiced frame detection fails completely. The reason for this is 2-fold. First, using CS preprocessing for this song, the ground truth that was build consists of 465 frames labeled as voiced dominant and only 15 frames labeled as unvoiced dominant. This comes from the fact, that too much energy is preserved in the center signal and the presented criteria to identify unvoiced dominant frames based on energy comparison almost fails. Additionally, the detection of the presence of a singing voice f_0, performed by the voiced singing stages, shows a very low accuracy ($< 35\%$) for this song. Hence, detecting unvoiced dominant frames by combining the LP error with the presence of a singing voice f_0 is unable to map the ground truth.

This leads to a mean accuracy improvement of 13.5% in using the panning index preprocessing. If the VAR is increased to 5dB, the UVFD accuracy difference between PI and CS decreases to mean value of 2.8%. This behavior is expected, since the unvoiced singing components are more and more dominating the mix and the ability to damp the accompaniment becomes less important. Thus, the higher the VAR the clearer unvoiced dominant frames can be distinguished from voiced dominant frames.

Song ID	**UVFD Accuracy**		
	PI	**CS**	**PI-CS**
1	86.6	83.1	3.5
2	43.0	46.2	-3.2
3	88.6	82.6	6.0
4	79.4	88.8	-9.4
5	72.0	76.1	-4.1
6	**88.3**	**0**	**88.3**
median	83	82.6	1.6
mean	76.3	62.8	13.5
CI	$\pm$15.4	$\pm$30.0	-14.6

Table 10: Unvoiced Dominant Frame Detection Accuracy comparison for different preprocessing strategies, Panning Index Preprocessing (PI), Center Signal Preprocessing (CS), all values in %, $VAR = 0$dB, 6-fold cross-validation

4.3.2 TFU Classification

The metrics used here are based on the following confusion matrix:

	predicted class	
	voice	music
ground truth		
voice	TP	FN
music	FP	TN

Table 11: Confusion matrix for Time-Frequency-Unit Classification

Table 12 presents the TFU Classification Accuracy along with the UVFD Accuracy. The achieved median classification accuracy reaches 69%, which represents a good result. In addition, it can be observed that the 3rd song exhibits a low classification accuracy in particular. This is due to the fact, that only a single SVM model is used to classify all TFU's, regardless from which gamma-tone filter output (auditory channel) a TFU originates. Experimental tests revealed, that the feature value distributions are not consistent over all channels. This decreases the classification accuracy for this song in particular, but also prevents higher accuracies for the whole set. This suggests, having separate models for each auditory channel would increase the overall accuracy.

Song	**TFU Classification**							
ID	**TP**	**FP**	**TN**	**FN**	**P**	**R**	**F**	**Acc.**
1	43.1	0	100	56.9	100	43.1	60.2	71.5
2	36.7	0	100	63.3	100	36.7	53.7	68.4
3	44.1	53.2	46.9	56.0	41.0	44.0	42.4	45.4
4	55.7	16.7	83.3	44.3	85.2	55.7	67.4	69.5
5	79.9	32.5	67.5	20.1	34.9	79.9	48.6	73.7
6	77.9	47.1	52.9	22.2	49.8	77.9	60.7	65.4
median	49.9	24.6	75.4	50.2	67.5	49.9	57.0	69.0
mean	56.2	24.9	75.1	43.8	68.5	56.2	55.5	65.7
CI	$\pm$16.3	$\pm$20.2	$\pm$20.2	$\pm$16.3	$\pm$26.3	$\pm$16.3	$\pm$8.0	$\pm$9.0

Table 12: Achieved accuracies for Time-Frequency-Unit Classification, all values in %, $VAR = 0$dB, 6-fold cross-validation, using panning index preprocessing

Next, the dependency of the classification accuracy on the preprocessing strategy is presented. Table 13 shows the comparison between PI preprocessing and CS preprocessing. Using the proposed PI preprocessing a median accuracy improvement of 9.8% is achieved. The significant increase in accuracy in general and for the songs 1 and 5 in

particular all arise from the same fact, which is the ability of the PI processing to suppress the accompaniment. Since the ground truth is build by comparing the occurring energies in each TFU of unvoiced singing and the premix without the unvoiced singing, the less energy is preserved from the accompaniment the more TFU's are labeled as class voice. If PI processing is used, 2251 TFU's are labeled as class *voice* and for CS processing it decreases to 1171, which represents a reduction by ∼50%. In addition, not only the instances for the class voice are reduced but the overall amount of training instances decreases by 35% as well, since less frames are labeled unvoiced dominant. This significant reduction in training instances makes the classification task inevitably more difficult.

Song ID	TFU Classification Acc.		
	PI	CS	PI-CS
1	71.5	50.0	21.5
2	68.4	64.5	3.9
3	45.4	54.2	-8.8
4	69.5	63.9	5.6
5	73.7	59.5	14.2
6	65.4	58.9	6.5
median	69.0	59.2	9.8
mean	65.7	58.5	7.2
CI	±9.0	±4.9	4.1

Table 13: Time-Frequency-Unit Classification Accuracy comparison for different preprocessing strategies, Panning Index Preprocessing (PI), Center Signal Preprocessing (CS), all values in %, $VAR = 0$dB, 6-fold cross-validation

Finally, to put the classification accuracies in perspective, we would like to stress the fact that this data set consists only of 17 seconds of audio recordings. Increasing the data set would very likely increase the classification accuracy. Additionally, the computational complexity was drastically reduced in contrast to [HJT08]. The authors proposed 36 MFCC values for 128 gamma-tone filters which would have resulted in more then $800 \cdot 10^3$ training instances for our training set, whereas we decreased the training instances to only ∼5000. Unfortunately, a comparison in classification accuracy to [HJT08] can not be presented here since, neither their algorithm, nor their training set is available to us.

4.4 Overall

To evaluate the overall singing voice separation quality, we use the BSS_EVAL Toolbox [4] for MATLAB®. This toolbox has gained a widespread use over the last years and is used to measure the performance of various source separation algorithms. For a detailed review the reader is referred to [VGF06]. The principle is to decompose a given estimate $\hat{s}(t)$ as sum of signals with allowed deformations

$$\hat{s}(t) = s_{target}(t) + e_{interf}(t) + e_{noise}(t) + e_{artif}(t) \tag{57}$$

where $s_{target}(t)$ represents the target signal, $e_{interf}(t)$ accounts for the interferences of the unwanted sources, $e_{noise}(t)$ represents perturbating noise, and $e_{artif}(t)$ is an „artifact“ term that may correspond to artifacts of the separation algorithm. Having the decomposed estimate, the following measures are calculated

Source to Distortion Ratio (SDR)

$$\text{SDR} =: 10log_{10}\frac{||s_{target}||^2}{||e_{interf} + e_{noise} + e_{artif}||^2} \tag{58}$$

Source to Interferences Ratio (SIR)

$$\text{SIR} =: 10log_{10}\frac{||s_{target}||^2}{||e_{interf}||^2} \tag{59}$$

Sources to Artifacts Ratio (SAR)

$$\text{SAR} =: 10log_{10}\frac{||s_{target} + e_{interf} + e_{noise}||^2}{||e_{artif}||^2} \tag{60}$$

Before we present the results, we would like to point out that the BSS_EVAL toolbox is not designed to handle stereo signals therefore, all signals are converted to center signals using Eq. 42.

Table 14 shows the resulting performance measures. Leaving song 2 aside, two observations can be made. First, all SDR and SAR values are negative, which indicates a high amount of artifacts. Since the proposed system is designed to extract the singing voice

4. Version 2.0, C. Févotte, R. Gribonval and E. Vincent, BSS EVAL Toolbox User Guide, IRISA Technical Report 1706, Rennes, France, April 2005. http://www.irisa.fr/metiss/bss eval/

Song ID	SDR (dB)	SIR (dB)	SAR (dB)
1	-6.48	36.12	-6.48
2	4.69	34.65	4.70
3	-1.02	36.48	-1.02
4	-2.49	30.20	-2.48
5	-2.95	43.63	-2.95
6	-5.99	29.30	-5.98
mean	-2.37	35.06	-2.37

Table 14: Evaluation of the proposed method for singing voice extraction using the BSS_EVAL Toolbox

signal with a high precision in contrast to a high recall, all singing voice components that are not extracted will be considered artifacts, after decomposition. Hence, the resulting low values. Second, for all songs a very high SIR is achieved. This is a very good result, since state of the art algorithms usually reside in the range of 5-30dB. Nevertheless, the SIR has to be seen in relationship to the SDR and SAR. For Song 2 all the results are in a desirable range and prove the potential of the proposed system.

To elaborate on the computational efficiency, we presented in table 15 the runtime for the task of singing voice extraction, listed for each file of the stereo data set (page 41). The test system was a MacBookPro 2.2GHz, Intel Core i7, 8 GB DDR3 RAM, MacOSX 10.6.8, MATLAB© R2009a (7.8.0.347).

Additionally, the time to train all classifiers including building the ground truth is presented in table 16. Since, especially in the case of grid-search and cross-validation the needed time to train the classifiers heavily depends on the parameter increments used for the grid search, the chosen grid-search represents a typical medium grid ($7 < ld(C) < 19; -5 < ld(\gamma) < 8$). More precise grids could easily increase the runtime by factor of 2 and higher.

Song ID	duration (sec)	execution time (sec)
1	2.2	27
2	2	21
3	3.5	40
4	4.5	64
5	2.7	40
6	5.8	77
mean	3.5	44.8

Table 15: Singing Voice Extraction execution time for stereo data set

	voiced singing (min)	unvoiced singing (min)
training	2.4	28.1
grid-search/cross-validation	3.5	70.1
total	5.9	98.2

Table 16: Singing Voice Extraction training runtime for stereo data set

5 Conclusion

A framework to extract singing voice signals from 2-channel polyphonic popular music recordings has been presented. Results show that the extraction of voiced singing voice based vocal melody estimation and a sinusoidal model achieves good separation results. The extraction of the unvoiced component needs to be further investigated.

The evaluation of individual processing stages and corresponding results have been presented separately. The panning index preprocessing increases the accuracy of the voiced singing classification by 12% in average and the unvoiced singing classification by 7.2% (median: 9.8%). The frame-wise pitch estimation yields 88.6% (median: 91.2%) on average for the presented stereo dataset. The presented partial tracker achieves a absolute mean accuracy of 85.7% (median 88.4%), which signifies that only 2.9% of the vocal pitches are discarded in the tracking process. In contrast to the pitch tracker proposed by Rahimzadeh [Rah09], this represents an improvement of 7%. The pitch track classification accuracy yields a mean value of 69.4% (median: 69%) on the stereo dataset and 87.8% (median: 90.4%) on 9 songs from the MIREX 2005 training set. The latter represents a mean improvement of 16.3% as opposed to Rahimzadeh.

The unvoiced dominant frame detection, based on linear Prediction and vocal f_0 presence, achieves 76.3% average accuracy. Panning Index preprocessing improves this frame detection task by 13.5%. The necessary training instances for the unvoiced time-frequency-unit classification are reduced by a factor of $1.6 \cdot 10^2$ and the classification accuracy reaches a mean value of 65.7% (median: 69%). The overall evaluation of the proposed method for the task of singing voice extraction yields a average Source to Interferences Ratio (SIR) of 35.1dB, while for every song of the set the SIR's are above 29dB. The average Source to Distortion Ratio (SDR) and Sources to Artifacts Ratio (SAR) is -2.4dB. Weaknesses of the proposed method have been discussed and suggestions for further improvements will be presented in the following chapter.

6 Discussion and Outlook

Maybe the most essential necessity, before considering any improvements to the proposed method, lies in expanding the dataset. Like for all supervised learning algorithms the size and content of the training dataset is of the essence. Hence, increasing the amount of training instances would certainly lead to higher classification accuracies and provide a better generalization.

As presented, the use of Panning Index preprocessing proved to increase the overall performance by preserving the singing voice signal and damping the accompaniment signal. This benefit could be further improved if the actual position of the singer in the stereo panorama is tracked, rather then assuming it to be constantly in the center. The tracking of the position could be realized by a learning algorithm, either unsupervised or supervised. This would allow the use of narrower windows for the spatial extraction of singing voice and therefore increase the damping of the accompaniment signal. Hence, the subsequent processing of voiced singing and unvoiced singing would benefit from this result.

The performance of the voiced singing processing stages could be improved in multiple ways. First, by increasing the amount of vocal pitch candidates and second by identifying the underlying source of a particular partial. As evaluation results of the multi-pitch estimation showed, increasing the number of pitch estimates per frame only increases the amount of detected true vocal pitches up to a certain point. This suggests that the limiting factor is likely to be the frequency resolution, i.e. the discriminability of occurring peaks in the magnitude spectrum. Methods like Time-Frequency Reassignment [HW01] or Multi-Resolution FFT [Dre06] could be considered to increase the effective frequency resolution. On the other hand, to be able to map a particular partial to its source or fundamental frequency would not only increase the pitch track classification accuracy (and thus the significance of the feature set), but the re-synthesis of the voiced singing voice as well. Specific partials could then be discarded from a pitch track if belonging to different sources or excluded in the singing voice re-synthesis process.

Another way to increase the pitch track classification accuracy is to maximize the significance of the derived ground truth. As presented, the ground truth class assignment of a pitch track depends on the amount of overlap with the reference vocal pitch trajectory. Hence, maximizing the overlap maximizes the significance of the resulting feature values. The proposed onset detection proved to be helpful in this task, which suggests that detecting onsets more precisely would be beneficial. On way could be to combine

the onset detection with a tempo estimation method, such that occurring onsets would reflect the rhythmic structure of a musical piece in a more accurate manner.

The detection and extraction of the unvoiced singing voice components could be improved mainly in two ways. First, in defining the ground truth more precisely, and second in introducing separate classifiers for each auditory channel. The former could be realized by hand labeling unvoiced singing components, which would allow the investigation of more suitable descriptive parameters for the unvoiced dominant frame detection. Furthermore, having separate classifiers for each auditory channel would very likely increase the classification accuracy, since it has been observed that the feature distributions are not consistent over all channels.

Finally, as for many music information retrieval algorithms, introducing a musicological model can improve the overall performance by providing a framework to put the analysis and synthesis of musical content into perspective, so to speak.

7 References

[Ave03] C. Avendano, "Frequency-Domain Source Identification And Manipulation In Stereo Mixes For Enhancement, Suppression And Re-Panning Applications," *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pp. 55–58, 2003.

[BL04] D. Barry and B. Lawlor, "Sound Source Separation: Azimuth Discriminiation and Resynthesis," *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFX-04)*, 2004.

[CL] C. Chang and C. Lin, "Libsvm: a library for support vector machines." [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[CL08] M. Cobos and J. López, "Singing Voice Separation Combining Panning Information And Pitch Tracking," *AES Convention Paper 7397*, 2008.

[CR08] A. Chanrungutai and C. Ratanamahatana, "Singing Voice Separation For Mono-Channel Music Using Non-negative Matrix Factorization," *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*, pp. 243–246, Oct. 2008.

[CSD10] E. Cano, G. Schuller, and C. Dittmar, "Exploring Phase Information In Sound Source Separation Applications," *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10), Graz, Austria*, pp. 1–7, Jun 2010.

[Dre06] K. Dressler, "Sinusoidal extraction using an efficient implementation of a multi-resolution FFT," *Proceedings of Ninth International Conference on . . .*, 2006.

[Fit10] D. FitzGerald, "Harmonic/Percussive Separation Using Median Filtering," *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pp. 1–4, Sep 2010.

[GOO06] M. M. GOODWIN, "Frequency-Domain Algorithms For Audio Signal Enhancement Based On Transient Modification," *AES Journal*, vol. 54, no. 9, pp. 1–14, 2006.

[HCL03] C. Hsu, C. Chang, and C. Lin. . . , "A Practical Guide To Support Vector Classification," Jan 2003.

[HJT08] C. Hsu, J. Jang, and T. Tsai, "Separation Of Singing Voice From Music Accompaniment With Unvoiced Sounds Reconstruction For Monaural Recordings," *Convention Paper 125th AES Convention, San Francisco*, pp. 1–6, Jul 2008.

[HW01] S. W. Hainsworth and P. J. Wolfe, "'Time-Frequency Reassignment for Music Analysis'," *Proc. International Computer Music Conference*, 2001.

[KD06] A. Klapuri and M. Davy, *Signal Processing Methods For Music Transcription*. Springer-Verlag New York Inc, 2006.

[Kla08] A. Klapuri, "Multipitch Analysis Of Polyphonic Music And Speech Signals Using An Auditory Model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 255–266, 2008.

[LW07] Y. Li and D. Wang;, "Separation Of Singing Voice From Music Accompaniment For Monaural Recordings," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1475 – 1487, May 2007.

[Mel91] D. K. Mellinger, "Event Formation And Separation In Musical Sound," *Ph.D. thesis, Stanford University, Department of Computer Science*, 1991.

[MQ86] R. McAulay and T. Quatieri, "Speech Analysis/Synthesis Based On A Sinusoidal Representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[OPB07] A. Ozerov, P. Philippe, and F. Bimbot, "Adaptation Of Bayesian Models For Single-Channel Source Separation And Its Application To Voice/Music Separation In Popular Songs," *Audio*, 2007.

[OPGB05] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbor, "One Microphone Singing Voice Separation Using Source-Adapted Models," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.* IEEE, 2005, pp. 90–93.

[P.04] S. P., "Non-negative Matrix Factor Deconvolution; Extraction Of Multiple Sound Sources From Monophonic Inputs," in *In Proc. 5th International Conference on Independent Component Analysis and Blind Signal Separation, Granada, Spain September 22â24*, 2004.

[Rah09] A. Rahimzadeh, "Detection Of Singing Voice Signals In Popular Music Recordings," *Diploma Thesis IEM*, pp. 1–79, Nov 2009.

[Rao09] P. Rao, "Musical Information Extraction From The Singing Voice," *ee.iitb.ac.in*, Nov. 2009.

[Rie09] S. Rieck, "Implementierung einer Ausfallsverschleierung mittels spektraler Signalmodellierung," Apr 2009, student project at the Institute of Electronic Music and Acoustics, University of Music and Performing Arts Graz.

[RP09] L. Regnier and G. Peeters, "Singing Voice Detection In Music Tracks Using Direct Voice Vibrato Detection," *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1685–1688, Apr. 2009.

[RVPK08] M. Ryynänen, T. Virtanen, J. Paulus, and A. Klapuri, "Accompaniment Separation And Karaoke Application Based On Automatic Melody Transcription," *2008 IEEE International Conference on Multimedia & Expo*, Jan 2008.

[SAP10] S. Sofianos, A. Ariyaeeinia, and R. Polfreman, "Singing Voice Separation Based On Non-Vocal Independent Component Subtraction And Amplitude Discrimination," *In: Proceedings of the 13th International Conference on Digital Audio Effects,(DARx 2010)*, 2010.

[Ser97] X. Serra, "Musical Sound Modeling With Sinusoids Plus Noise," *Musical signal processing*, pp. 497–510, 1997.

[Ter98] E. Terhardt, *"Akustische Kommunikation, Grundlagen mit Hoerbeispielen"*, 1998.

[VB05] S. Vembu and S. Baumann, "Separation Of Vocals From Polyphonic Audio Recordings," *Proc. ISMIR*, pp. 337–344, Nov. 2005.

[VGF06] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[ZBS02] D. Zangger Borch and J. Sundberg, "Spectral Distribution Of Solo Voice And Accompaniment In Pop Music," *speech.kth.se*, Sep. 2002.

List of Figures

List of Tables

Appendix A

Average Fisher's Ratio

Feat. No.	FR	included	Feat. No.	FR	included
1	2.4294	x	27	2.8985	x
2	1.702	x	28	2.8963	x
3	0.035822		29	0.0043187	
4	0.37023	x	30	0.0033997	
5	0.37144	x	31	0.0033997	
6	0.36715	x	32	0.16209	x
7	0.3692	x	33	0.21001	x
8	0.37404	x	34	0.14168	x
9	0.29172	x	35	1.0562	x
10	0.37061	x	36	0.031896	
11	0.026523		37	0.053269	
12	0.027551		38	0.10983	x
13	0.02502		39	0.11728	x
14	0.39862	x	40	0.15193	x
15	0.43488	x	41	0.080696	
16	0.30686	x	42	0.0060718	
17	0.0018045		43	0.023234	
18	0.46457	x	44	0.086177	
19	0.46339	x	45	0.048238	
20	0.0014838		46	0.19057	x
21	0.0035455		47	0.14174	x
22	0.0074504		48	0.16576	x
23	0.07212		49	0.040649	
24	1.5785	x	50	0.0015568	
25	2.7387	x	51	0.0010712	
26	3.3152	x			

Table 17: Average Fisher's Ratios (FR) for each feature, features that are included in the final subset used in Pitch Track Classification are marked with x