Audio Engineering Project

Real-time capable analysis and visualization of bell ringing

Felix Holzmüller
Matr. No.: 01573159

Electrical Engineering and Audio Engineering, Master's Program - UF 066 413
University of Music and Performing Arts Graz

Supervisor: Univ.Prof. Dipl.-Ing. Dr.techn. Alois Sontacchi

Graz, January 9, 2022

institute of electronic music and acoustics

Abstract

Bell-ringing is a fundamental part of ecclesiastical rites. The aim of this interdisciplinary project involving the long night of churches and the Akademie Graz is to provide a visualization of bell ringing, aimed at hearing impaired persons. In a first step, an analysis algorithm is created. Therefore, spectral and temporal features are analyzed. Different calculation techniques for activations, based on frequency-domain analysis and a non-negative matrix factorization (NMF) are derived. A model for local consonance is presented. In a next step, a real-time analyzation and visualization based upon the found parameters is created.

Zusammenfassung

Glockenläuten ist ein grundlegender Bestandteil kirchlicher Kultur. Im Zuge dieses interdisziplinären Projekts soll in Kooperation mit der Langen Nacht der Kirchen und der Akademie Graz eine Visualisierung kreiert werden, um auch gehörlosen Menschen diese Tradition zugänglich zu machen. Dabei soll in einem ersten Schritt eine Analysemethode für Geläut gefunden werden. Besonderer Augenmerk liegt hier auf spektralen und zeitlichen Features wie etwa Grundfrequenz, Obertonstruktur, rhythmische Muster und dynamische Verläufe. Verschiedene Ansätze zur Berechnung der Aktivierungen, basierend auf einer Zeit-Frequenzanalyse und einen non-negative matrix factorization (NMF) werden hergeleitet. Ein Modell zur Bestimmung der wahrgenommenen Konsonanz wird präsentiert. Mit diesen Parametern wird anschließend in Echtzeit eine Analyse und Visualisierung vorgenommen.

Contents

Chapter 1

Introduction

The soundscape in cities and villages is for normal hearing persons an ordinary sensation, the individual sound sources are often not even perceived deliberately. What seems common to them, can be only an abstract concept for deaf or hearing impaired people. In the course of the interdisciplinary project “Kultur Inklusiv” [Aka21], a variety of accessible cultural projects in an urban context were carried out.
One element accompanying us in towns and villages are the sound of bells from bell towers and churches. In the course of this specific project [Kur+21, pp. 56–57], in cooperation with Akademie Graz, Kirchen Kultur Graz and Gehörlosenverband Steiermark among others, bell sounds were to be analyzed and visualized, especially for hearing impaired people. It was desired to stay as close to physically measurable and perceived parameters as possible in this project. The challenge lies in the abstraction and translation of a sound event into graphical domain, for which deaf people do not have an association for.

In chapter 2, the characteristics of bells in an acoustical context are shown. Chapter 3 deals with the principals of a constant-Q transform as basis for further analysis. A central item used for visualization are the temporal activations of different sound sources in chapter 4, where different calculation approaches are presented. In addition to the extracted physical parameters, the auditory/psycho acoustics related perceptual measure of consonance is treaded in chapter 5. Chapter 6 deals with the implementation of the analysis algorithm in a real-time capable environment, chapter 7 covers a first concept for visualization. The findings are summarized in chapter 8.

Chapter 2

Characteristics of bell ringing

For successfully finding an analyzation model, basic knowledge of the characteristics of the source is necessary. Even more in this case, as the related project is meant for deaf and hearing impaired people, who do not know the sound of a bell. So the essence of the sound characteristics has to be found using knowledge of the sound generation itself.

2.1 Construction and sound generation

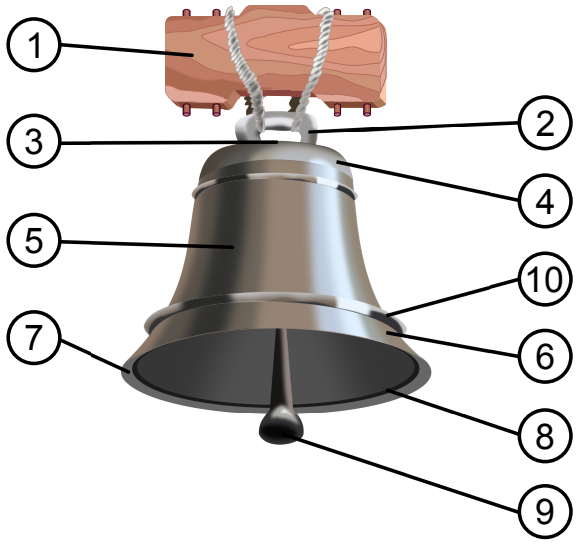


Figure 2.1 – Schematic structure of a bell [Wik20]

A schematic illustration of an ordinary bell is shown in fig. 2.1. It is either fixed in a static position or mounted to move freely at the bell yoke ① via the canons ②. The typical form is defined by the crown ③, the closed upper boundary, and the curvature with shoulder ④, waist ⑤ and sound bow ⑥ ending at the lip ⑦, which surrounds the mouth ⑧. Waist and sound bow are separated by the bead line ⑩. Fixed bells are typically excited from

the outside using a hammer, while moveable bells are swung so that the clapper ⑨ at the inside hits the bell.

First systematic experiments regarding vibrational patterns and partial tones of bells have ben made by Chladni [Chl87], Helmholtz [Hel63] and Rayleigh [Ray90]. The radiated sound can be described by the oscillation of the bell. The vibrational pattern can be decomposed in different normal modes. Those are motion patterns, where all parts move sinusoidally with a fixed phase relation and nodes at a certain frequency. An example for those normal modes can be seen in fig. 2.2, where dark areas denote nodes and white areas the maxima of the movement. The strength and damping of modes, which result in partials with different amplitudes and decay times, are defined by the geometry of the bell. The excitation can be assumed as impulse-like, which results in an excitation of all normal modes. For antisymmetric modes, the so called mode splitting can occur. Antisymmetric normal modes can be further decomposed in two movement patterns. Ideally, both patterns should have the same resonance frequency. However, due to production variations and therefore a not perfectly rotational symmetric bell, modes occur at slightly different frequencies, resulting in a noticeable beat tone [Fle97]. The acoustical properties will be discussed more in detail in section 2.2.

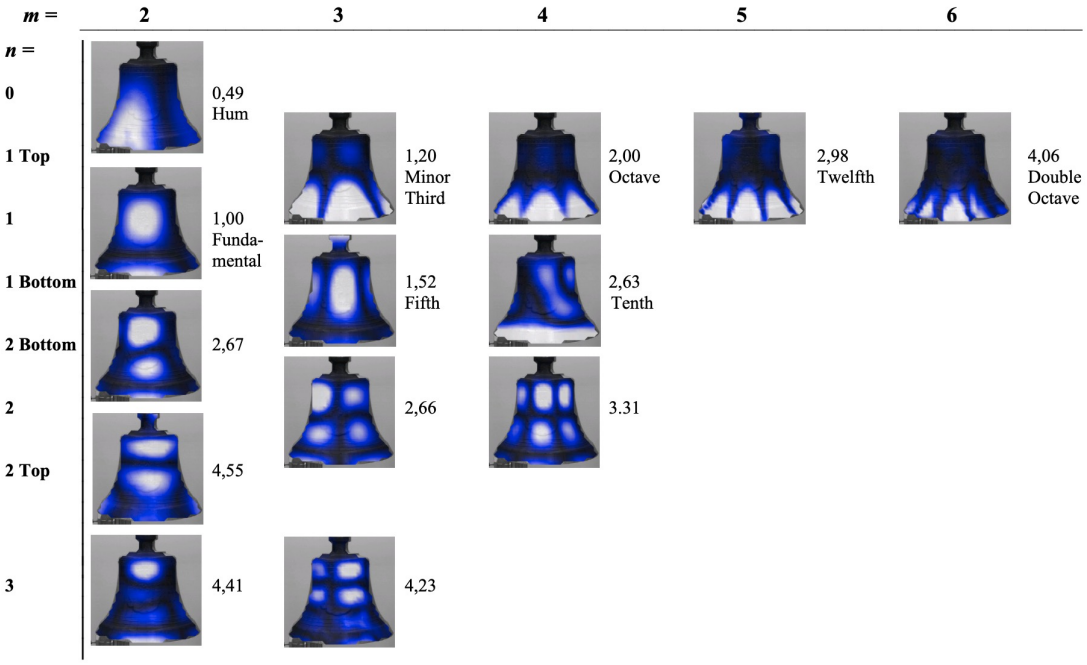


Figure 2.2 – Exemplary movement patterns and frequency ratios of a bell. [Fle97, p. 102]

The strike tone (see section 2.3), the perceived pitch of the bell, is mainly influenced by size and weight of the bell. Major impact on the timbre has the geometry of the bell as well as precise, symmetric manufacturing [Fle97]. A more detailed explanation about constructional details and its impact on the timbre can be found e.g. in [Fle97; Wer04].

2.2 Harmonic structure

The typical bell sound can be decomposed in a series of inharmonic partials. This means that in contrary to sound from the majority of musical instruments, the frequencies of the overtones are not distributed in whole-number ratios of the fundamental frequency. In extensive investigations, about 30 partial tones can be detected [FFS07]. An exemplary spectrogram and the more detailed analysis results can be seen in fig. 2.3 and table 2.1. Due to historical reasons, some dominant partials have their own denotation (cf. fig. 2.2). The second partial is defined as fundamental frequency, the partial an octave below is called hum tone. The higher, dominant partial are named after the corresponding musical intervals (e.g. Third, Fifth). Especially the Fifth and the Tenth are weak and nearly inperceivable partials [Wer04, p. 11].

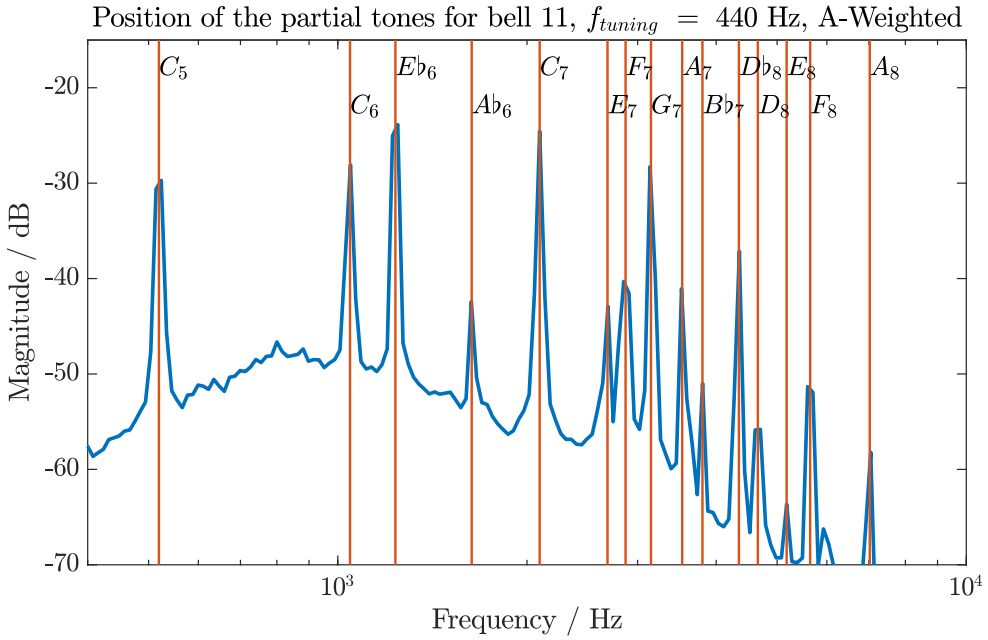


Figure 2.3 – Spectrogram of a bell sound from Graz Mausoläum.

Another quite unique feature of bell sounds are the drastically different decay times of the partials. While especially the hum tone and the fundamental tone can have decay times in the range of two- or even three-digit seconds, some partials with similar amplitudes have decay times of well below $T_{60} < 5\,\mathrm{s}$. So these partials can only be perceived directly after excitation, while the hum tone can be heard for minutes [FFS07].

Index	Frequency (Hz)	Notation	Detuning (cent)	Magnitude (dB)	Frequency-Ratio	Name
1	519.3	C_5	+13	-25.1	0.5	Hum
2	1045.8	C_6	+1	-28.2	1	Fundamental
3	1235.0	$E\flat_6$	+13	-22.1	1.18	Third
4	1633.2	$A\flat_6$	+29	-43.4	1.56	Fifth
5	2095.2	C_7	-2	-25.8	2	Octave
6	2686.2	E_7	-32	-44.1	2.57	Tenth
7	2870.2	F_7	-47	-41.1	2.74	
8	3149.8	G_7	-8	-28.9	3.01	Twelth
9	3531.7	A_7	-6	-42.0	3.38	
10	3803.7	$B\flat_7$	-34	-52.0	3.64	
11	4347.2	$D\flat_8$	35	-37.8	4.16	Double Octave
12	4659.3	D_8	+15	-55.3	4.46	
13	5178.1	E_8	+32	-64.1	4.95	
14	5644.2	F_8	-17	-49.5	5.4	
15	7021.1	A_8	+5	-57.2	6.71	

Table 2.1 – Detailed analysis of a bell sound from Graz Mausoläum. Notation and detuning are calculated for the Fundamental as reference (tuning frequency of $440\,\mathrm{Hz}$).

2.3 Human perception

When discussing bell sounds, also human perception and psychoacoustics should be taken into account. The perceived pitch after excitation, the so called strike tone, is one of the major characteristics of a bell. Due to the inharmonic partials, objective calculation can be quite extensive, normally it is determined perceptually by the manufacturer. As a rule of thumb, the strike tone of medium sized bells lies one octave below the frequency of the Octave, while for very big bells (with fundamentals below $\sim 200\,\mathrm{Hz}$), the strike tone lies two octaves below the Double Octave. Note that this procedure is needed, as the Octave and Double Octave usually are inharmonic to the fundamental frequency [Wer04, p. 22]. In the example from table 2.1, the strike tone would therefore lie one octave at about $1047.6\,\mathrm{Hz}$, slightly above the Fundamental.

It has been investigated experimentally, that only a smaller number of about 9 to 15 partials are necessary for perception of a typical bell sound. Especially the partials between hum tone and Double Octave are necessary for a convincing impression [FFS07].

Bell ringing is normally not reduced to a single bell. Instead, several bells begin to ring sequentially. As this is presumably done to avoid a resonance disaster of the bell tower, it has to be taken into account for human perception. While the excitations at the beginning with one ore two bells can be detected quite good, more sound components affect and impair this ability. So a transition from a well defined excitation pattern to a slightly pulsating sound mixture can be perceived.

Chapter 3

Constant-Q Transform (CQT)

As basis for a robust calculation of activations, a time-frequency representation of the input signal is needed, in this case the constant-Q transform (CQT). The CQT was originally introduced in 1978 by James Youngberg [YB78]. In 1991, it was used the first time in the context of musical analysis by Judith Brown [Bro91] and emerged since then as a valuable tool for music information retrieval. In contrary to the discrete Fourier transform (DFT), the calculated frequency components of the CQT are not spaced equally but logarithmically, based on a geometric series. Hence the frequency spacing corresponds well with the western music system and human perception to a certain degree and can be set e.g. to a quater-tone resolution. This results in a high frequency resolution for low frequencies and low frequency resolution for high frequency components. The time resolution behaves inversely. This property is implied at the CQT representation of an Kronecker delta in fig. 3.1.

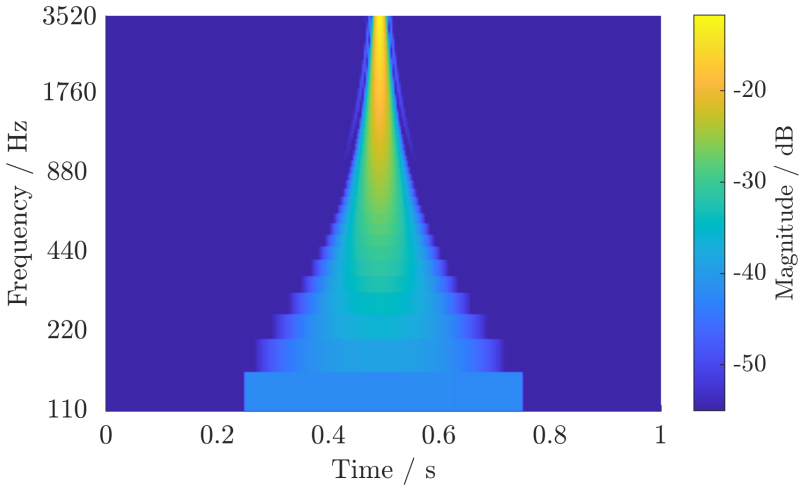


Figure 3.1 – CQT representation of an Kronecker delta.

In this chapter, the conventional, originally proposed calculations are presented as well as an effective real-time capable approach. Parts were also already published in [Hol+20a;

Hol+20b]. Additionally, it should be noted, that some approaches for an invertible CQT based on the theory of non-stationary Gabor frames were made, e.g. in [Vel+11; Hol+13; Sch+14]. These are not discussed in the following, as they have no relevance to this specific application.

3.1 Conventional CQT

At first, all necessary parameters for the transform will be defined. The CQT can be seen as filter bank of a signal with logarithmically spaced center frequencies f_k and a constant quality factor Q for all filters, where k denotes the index of the frequency bin. The center frequencies can be expressed as a geometric series

$$f_k = f_0 \, 2^{\frac{k}{b}} \tag{3.1}$$

with a minimal analysis frequency f_0 and the desired number of frequency bins per octave b.

For convenient usage it is quite practical to set the Q factor directly as a function of b. It can be expressed as

$$Q = \frac{f}{\delta f} = \frac{f}{\left(2^{\frac{1}{b}} - 1\right) f} = \left(2^{\frac{1}{b}} - 1\right)^{-1} \tag{3.2}$$

using an alternative definition via the bandwidth δf, so as a function only depending on of b. δ can be seen as frequency difference in percent between two bins, which is in a constant-Q case constant for all bins. Quite similar to the Q-factor, the *absolute* bandwidth B_k for each bin

$$B_k = f_k \left(2^{\frac{1}{b}} - 2^{\frac{1}{-b}}\right) \approx 2 \, \frac{f_k}{Q} \tag{3.3}$$

can be defined. This absolute bandwidth shall not be confused with the $-3\,\mathrm{dB}$ bandwidth.

In order to obtain a constant Q factor, the window size N_k for each bin and therefore the number of analyzed samples decreases inproportionaly to f_k. It can be obtained as

$$N_k = \left\lceil \frac{f_s}{f_k} \, Q \right\rceil \tag{3.4}$$

with the sample rate f_s of the signal.

Now that all parameters for the CQT are prepared, the transform itself can be addressed.

The CQT is simply defined as normalized DFT with the different window lengths N_k for each frequency bin. Hence the transform $X[k]$ of a signal $x[n]$ can be calculated as

$$X[k] = \frac{1}{N_k} \sum_{n=0}^{N_k-1} g_k[n]\, x[n]\, e^{-j2\pi Qn/N_k} \tag{3.5}$$

with a window $g_k[n]$. The window function has the same shape for each component. Traditionally a Hamming or Hann window

$$\begin{aligned} g_{k,\text{Hamming}}[n] &= a + (1-a)\cos\left(2\pi n/N_k\right), \quad \text{with } a = 25/46 \\ g_{k,\text{Hann}}[n] &= \cos^2\left(\pi n/N_k\right) \end{aligned} \tag{3.6}$$

is used to avoid leakage into adjacent frequency components.

3.1.1 Comparison to the discrete Fourier transform (DFT)

Although the CQT is obviously related to the DFT, some major differences can be observed. The most mentionable differences according to [Bro91] can be seen in table 3.1.

	DFT	CQT
Bin frequencies f_k	$k \cdot b$	$f_0\, 2^{\frac{k}{b}}$
Window length N	constant	variable: $N_k = \frac{f_s Q}{f_k}$
Frequency resolution Δf	constant: $\frac{f_s}{N}$	variable: $\frac{f_k}{Q}$
Quality factor Q	variable: $\propto k$	constant
Kernel	$e^{-j2\pi kn/N}$	$e^{-j2\pi Qn/N_k}$

Table 3.1 – Comparison of DFT and CQT [Bro91].

3.2 Efficient implementation

The direct calculation of CQT coefficients as proposed [Bro91] is computationally quite extensive. Due to the varying window lengths and therefore lack of symmetries, efficient algorithms such as the FFT cannot be used directly. Nonetheless, the FFT can be the basis for a more advanced algorithm as proposed in [BP92], [Vel+11] and [Sch+14]. For the sake of consistency and in view of the implementation, the algorithm and notation of [Hol+20a] is used in the following.

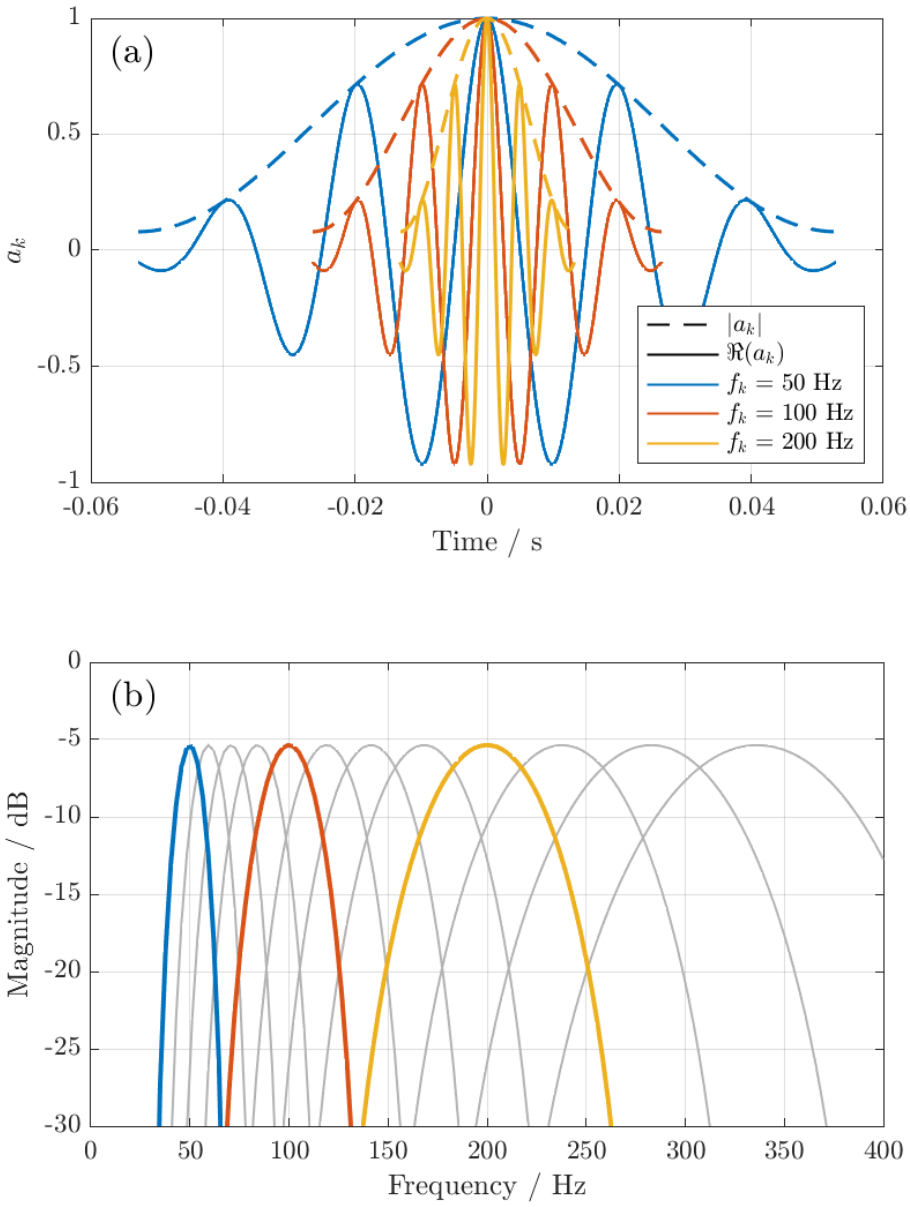


Figure 3.2 – Exemplary visualization of the atoms a_k with $f_0 = 50\,\text{Hz}$, $b = 4$ and $f_s = 44.1\,\text{kHz}$ in
(a) time-domain
(b) frequency-domain.

At first, eq. (3.5) can be rewritten as

$$X[k,n] = \sum_{m=0}^{N_k-1} x[m]\, a_k^*[m-n], \qquad n,k \in \mathbb{N}, \text{ with} \tag{3.7}$$

$$a_k[m] = g_k[m]\, e^{j2\pi m \frac{f_k}{f_s}}, \qquad m \in \mathbb{Z}. \tag{3.8}$$

In this form, the window function g_k and the kernel are combined to the complex conjugated localization functions (window functions) $a_k^*[m]$, the so called *atoms*. An exemplary visualization of a_k in time- and frequency-domain is shown in fig. 3.2.

3.2.1 Calculating the CQT using fast convolution

When looking at eq. (3.7), one can see the resemblance to a convolution, although the time indexing of $a_k^*[m]$ is reversed. When assuming g_k to be a zero-centered, symmetric function, the atoms can be rewritten to $a_k^*[m] = a_k[-m]$. So the CQT computation can be rearranged to

$$\begin{aligned} X[k,n] &= \sum_{m=0}^{N_k-1} x[m]\, a_k^*[m-n] \\ &= \sum_{m=0}^{N_k-1} x[m]\, a_k[n-m] \\ &= (x * a_k)[n]. \end{aligned} \tag{3.9}$$

The convolution can be efficiently computed by a *fast convolution*. That is, using the FFT to compute the convolution by means of a multiplication in the frequency domain and going back to time domain via an inverse-FFT (IDFT), defined as

$$\begin{aligned} X[k,n] &= (x * a_k)[n] \\ &= \mathcal{F}^{-1}_{i \mapsto n}\{[\mathcal{F}_{n \mapsto i}\{x\} \cdot \mathcal{F}_{n \mapsto i}\{a_k\}](i)\}[n]\,, \end{aligned} \tag{3.10}$$

where i shall denote a STFT-bin and k a CQT-bin. The use of the STFT requires block processing of the signal. The block lengths correspond to the DFT length N_{DFT}. The length of the STFT N_{STFT} depends on the maximum window length $N_{\text{max}} = \max(N_k)$, occurring at the lowest analysis frequency f_0. The signal shall be blocked with 50% overlap and windowed with a *Hann*-window before its transformation into the frequency domain. For a computationally more efficient implementation meaning less overlap, the usage of *Tukey*-windows as proposed in [Hol+13] would be also possible. When using

oversampling with a factor $os \in \mathbb{N}^+$ its length is defined as

$$\begin{aligned} N_{\text{STFT}} &= os \cdot \text{nextPower}_2\left(N_{max}\right) \\ &= os \cdot \text{nextPower}_2\left(Q\,\frac{f_s}{f_0}\right) \\ &= os \cdot 2^{\left\lceil \log_2\left(Q\frac{f_s}{f_0}\right)\right\rceil}. \end{aligned} \tag{3.11}$$

As the algorithm only deals with real valued input signals, we can optimize by only calculating and storing the STFT for positive frequencies.

There are two immediate optimizations for this procedure. Firstly, the input signal needs to be transformed to the frequency domain only once for the calculation of all frequency bins. Secondly, there is no need to repeatedly transform the localization functions, they can be designed and stored in the frequency domain prior to the transformation itself, where they constitute window functions A_k. This gives the advantage of a window with compact support, so that applying the window is computationally easy. Therefore, eq. (3.10) can be further simplified to

$$Y(i) = \mathcal{F}_{n\mapsto i}\{x[n]\}(i) \tag{3.12}$$

$$X[k,n] = \mathcal{F}^{-1}_{i\mapsto n}\{Y(i)A_k(i)\}[n]. \tag{3.13}$$

As a drawback, slight ripple can be observed due to the window's infinite support in time domain after transforming back into the CQT domain. A *Hann*-window is proposed for its good sidelobe suppression and narrow mainlobe as well as perfect overlapping.

3.2.2 Subsampling in the frequency domain

In this state the algorithm's output, namely one coefficient for each bin in each time step, contains a large amount of redundancy. The Shannon-Nyquist sampling theorem (in its extension to non-baseband signals) states that this redundancy can be removed by subsampling the output of each bin: as long as the sampling rate after subsampling f_s^k is at least the size of the absolute bandwidth B_k, no information will be lost.

$$f_s^k \geq B_k \tag{3.14}$$

To further reduce the computational effort it is also possible to perform the subsampling in the frequency domain. This is done by applying an inverse STFT (ISTFT) with $N_{\text{ISTFT}} < N_{\text{STFT}}$ only along the range where the respective frequency domain window is non-zero, or in other words, by shifting the windowed spectrum to the baseband before transforming back to the time domain with a lower resolution IDFT.

The lower and upper bounds in terms of STFT-bins $i_{u,k}$ and $i_{l,k}$ mark the position of upper and lower bounds of A_k as

$$i_{u,k} = \left\lfloor \left(f_k \cdot 2^{\frac{1}{b_k}} \right) \frac{N_{\text{STFT}}}{f_s} \right\rfloor, \tag{3.15}$$

$$i_{l,k} = \left\lceil \left(f_k \cdot 2^{\frac{-1}{b_k}} \right) \frac{N_{\text{STFT}}}{f_s} \right\rceil. \tag{3.16}$$

The values of the windows itself are obtained from a large, precalculated Hann window A_{lookup} of length M. For every sampling point of A_{lookup} a frequency $f_{A_{\text{lookup}}}[k, m]$ is assigned for each bin k, based on the CQT's logarithmical frequency spacing. These are calculated as

$$f_{A_{\text{lookup}}}[k, m] = f_k \cdot 2^{\frac{-\lfloor M/2 \rfloor + m}{\lfloor M/2 \rfloor \cdot b_{\text{new},k}}} \qquad \text{for } m = 0, 1, \ldots, M-1.$$

A length of $M = 8 \cdot N_{\text{ISTFT}}$ (see eq. 3.19) is more than sufficient for usage without further interpolation. The values of A_{lookup} whose corresponding frequencies $f_{A_{\text{lookup}}}[k, m]$ are closest to the frequencies of the DFT-bins $f_i = i \cdot \frac{N_{\text{STFT}}}{f_s}$ for $i_{l,k} \leq i \leq i_{u,k}$ are chosen and stored as the window function A_k for the k-th CQT-bin. This procedure is visualized in figure 3.3.

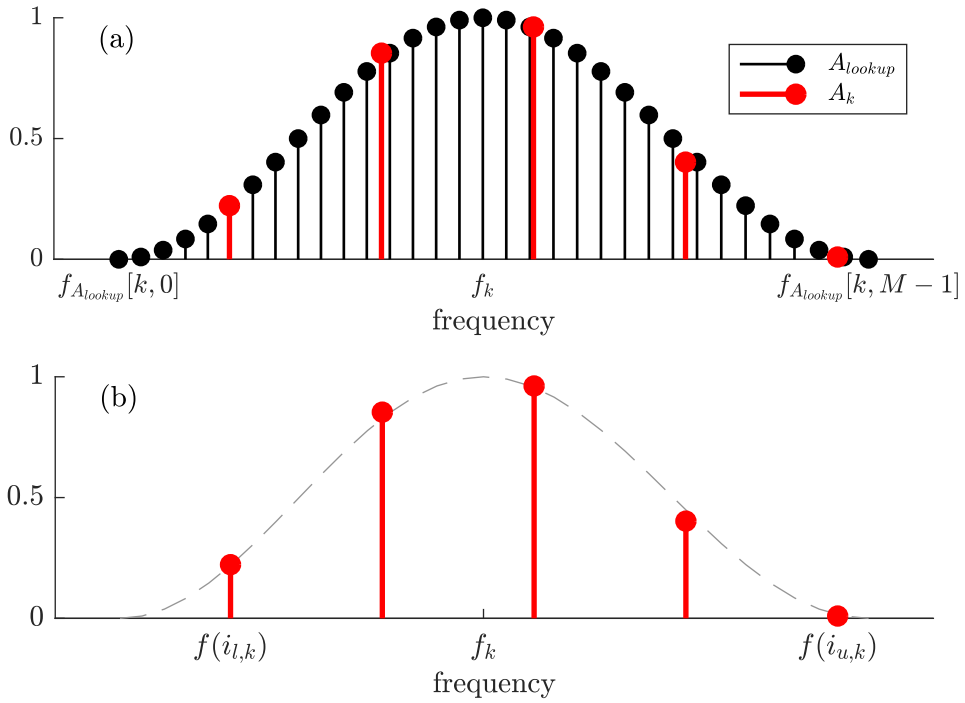


Figure 3.3 – Exemplary calculation of A_k.
(a) Obtaining coefficients from A_{lookup}.
(b) Exemplary visualization of A_k.

The shift to the baseband is computed with:

$$Y'(i,k) = \begin{cases} Y(i_{l,k}+i) \cdot A_k(i_{l,k}+i), & \text{for } 0 \leq i \leq i_{u,k} - i_{l,k} \\ 0, & \text{else} \end{cases} \tag{3.17}$$

The spectrum is then transformed inversely to obtain X' as

$$X'[k, n_k] = \mathcal{F}^{-1}_{i \mapsto n_k}\{Y'(i,k)\}. \tag{3.18}$$

X' is the subsampled CQT of x, with the time variables n_k of the individual channels progressing with f_s^k. If the ISTFT is set to the maximal needed size, a common time axis can be found, namely with

$$N_{\text{ISTFT}} = \text{nextPower}_2\left(N_{\text{STFT}} \frac{B_{k,\max}}{f_s}\right). \tag{3.19}$$

In this implementation the maximum required ISTFT length is used for all bins, applying zero-padding when necessary. Although slightly less efficient, this has the advantage of all channels running at the same rate. Finally, the time-CQT representation is obtained using an overlap-and-add algorithm on the absolute values $|X'[k, n_k]|$ using

$$hs = 2\,\frac{N_{\text{ISTFT}}}{os}. \tag{3.20}$$

as the hopsize for reconstruction of the blocks.

3.3 Enhancement of time resolution at low frequencies

Due to the relatively high Q factors, the bandwidth at low frequencies can be quite narrow, and therefore the time-resolution gets very low.

Example 1:
Common settings for music analysis are chosen with a minimal analysis frequency of $f_0 = 55\,\text{Hz}$, a sample rate of $f_s = 44.1\,\text{kHz}$ and a quarter-tone resolution with $b = 24\,\text{bins/oct}$. Following eq. (3.3), the absolute bandwidth at the lowest bin is given with

$$B_{k,\text{new}}\Big|_{k=0} = f_k \left(2^{\frac{1}{b}} - 2^{-\frac{1}{b}}\right)\Big|_{k=0} = 55\,\text{Hz} \cdot \left(2^{\frac{1}{24}} - 2^{-\frac{1}{24}}\right) \approx 3.18\,\text{Hz}.$$

A convenient measure for the time-resolution is the needed number of samples N_k.

According to eq. (3.4) and eq. (3.2), these can be calculated for the lowest bin as

$$N_k\Big|_{k=0} = \left\lceil \frac{f_s}{f_k}\, Q \right\rceil_{k=0} = \left\lceil \frac{44\,100\,\mathrm{Hz}}{55\,\mathrm{Hz}} \left(2^{\frac{1}{24}} - 1\right)^{-1} \right\rceil = 27\,364\,\mathrm{samples}$$

which equals approximately $0.620\,\mathrm{s}$.

This behavior can be problematic in applications, where the time structure is of importance. To overcome this drawback, the *constant*-Q case can be softened to a *variable*-Q case, where the absolute bandwidths are increased by

$$B_{k,\mathrm{new}} = B_k + \gamma = f_k \left(2^{\frac{1}{b}} - 2^{-\frac{1}{b}}\right) + \gamma \tag{3.21}$$

with a fixed amount $\gamma\ [\mathrm{Hz}] \in \mathbb{R}_{\geq 0}$ [Sch+14].

With the definition in eq. (3.3) and eq. (3.21), $Q_{k,\mathrm{new}}$ and $N_{k,\mathrm{new}}$ are frequency dependent and can be obtained as

$$Q_{k,\mathrm{new}} \approx 2\, \frac{f_k}{B_{k,\mathrm{new}}}, \tag{3.22}$$

$$N_{k,\mathrm{new}} = \left\lceil \frac{f_s}{f_k}\, Q_{k,\mathrm{new}} \right\rceil = \left\lceil 2\, \frac{f_s}{B_{k,\mathrm{new}}} \right\rceil. \tag{3.23}$$

Note that the center frequencies of the CQT-bins f_k are not affected by γ. The effect of γ on Q and B_k is shown in fig. 3.4.

Example 2:
Let's assume the same parameters as in example 1, but this time with $\gamma = 20\,\mathrm{Hz}$. The new bandwidth can be calculated with eq. (3.21) as

$$B_{k,\mathrm{new}}\Big|_{k=0} = B_k + \gamma\Big|_{k=0} \approx 3.18\,\mathrm{Hz} + 20\,\mathrm{Hz} = 23.18\,\mathrm{Hz}.$$

With eq. (3.22), the needed number of samples are

$$N_{k,\mathrm{new}}\Big|_{k=0} = \left\lceil 2\, \frac{f_s}{B_{k,\mathrm{new}}} \right\rceil = \left\lceil 2\, \frac{44\,100\,\mathrm{Hz}}{23.18\,\mathrm{Hz}} \right\rceil = 3806\,\mathrm{samples}$$

which corresponds to a time of approximately only $0.086\,\mathrm{s}$.

As one can see, $\gamma > 0$ has only a noticeable effect towards low frequencies, where the parameter is in the range or even bigger than B_k. The time-resolution can be drastically enhanced on the cost of frequency-resolution and the constant-Q property (therefore slightly more complex window computation). The effect on higher frequencies with $B_k \gg \gamma$ can be neglected. With increasing frequency, the constant-Q case is reached asymptotically. This properties also resembles the human perception and the theory of equivalent rectan-

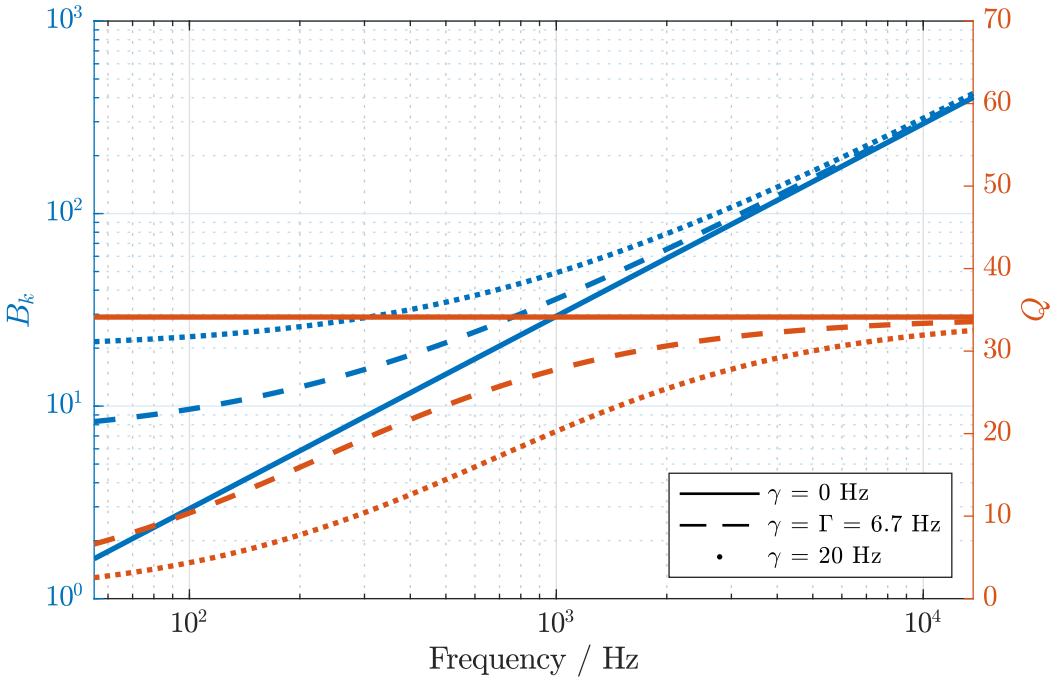


Figure 3.4 – Comparison of Q and B_k with the parameters of example 1 with $\gamma = 0\,\mathrm{Hz}$ (constant-Q), $\gamma = \Gamma = 6.7\,\mathrm{Hz}$ and $\gamma = 20\,\mathrm{Hz}$ (variable-Q).

gular bandwidths (ERB) [GM90], since the hearing resembles a constant-Q system only above approximately $500\,\mathrm{Hz}$. With a choice of

$$\gamma = \Gamma = \frac{24.7}{0.108} \cdot \frac{1}{Q}\,, \tag{3.24}$$

the bandwidths B_k are a constant fraction of the ERB critical bandwidth [Sch+14].

Chapter 4

Calculation of activations

The basic objective of the analyzation block is to gain as much musical information about the bell sound as possible. Two of the most important and striking parameters are the temporal and harmonic structure of the sound, to be more specific the activations/onsets over the course of time and the overtone structure for each bell. Ideally, the activations for each individual bell is obtained. As the harmonic information is a byproduct in many activation-detection algorithms, it is not descried in a separate chapter.

In the following, different approaches for the calculation of activations and onsets are explained. To anticipate the outcome, each algorithm has its advantages and disadvantages, so the choice depends on the specific situation.

4.1 Modified Kullback-Leibler divergence

As stated in [ANP11; Bro06], the modified Kullback-Leibler divergence presents a simple yet powerful measure to highlight musical (percussive) onsets and therefore the activation of bells. It is used to evaluate the distance between two consecutive spectral vectors. It emphasizes large positive energy changes while inhibiting small changes as well as decays. As a result, quite sharp peaks at percussive onsets can be observed.

As the name suggests, the measure is based on the Kullback-Leibler distance

$$D_{\mathrm{KL}}[n] = \sum_{k=0}^{K-1} |X[k,n]| \ \ln\left(\frac{|X[k,n]|}{|X[k,n-1]|}\right) \tag{4.1}$$

where $X[k,n]$ is a spectral vector of length K, e.g. the CQT coefficients at time n with the frequency bins $k \in [0; K-1]$.

With the definition in eq. (4.1), two obvious problems arise. For one, the expression can get undefined for a series of small elements, hence a regularization can be introduced. Secondly, negative values can be reached which can be problematic for peak-picking algorithms and would increase their complexity. Also the $|X[k,n]|$ weighting can be removed to be independent of amplitude changes [HM03]. When addressing all these issues, the modified Kullback-Leibler divergence is redefined to

$$D_{\mathrm{mKL}}[n] = \sum_{k=0}^{K-1} \ln\left(1 + \frac{|X[k,n]|}{|X[k,n-1]| + \epsilon}\right) \tag{4.2}$$

with the small additional regularization parameter ϵ to avoid large variations at very low amplitude levels. A higher value for ϵ results typically in a smoothing of the activation curve. An example and comparison of the normal and modified Kullback-Leibler divergence can be seen in fig. 4.1.

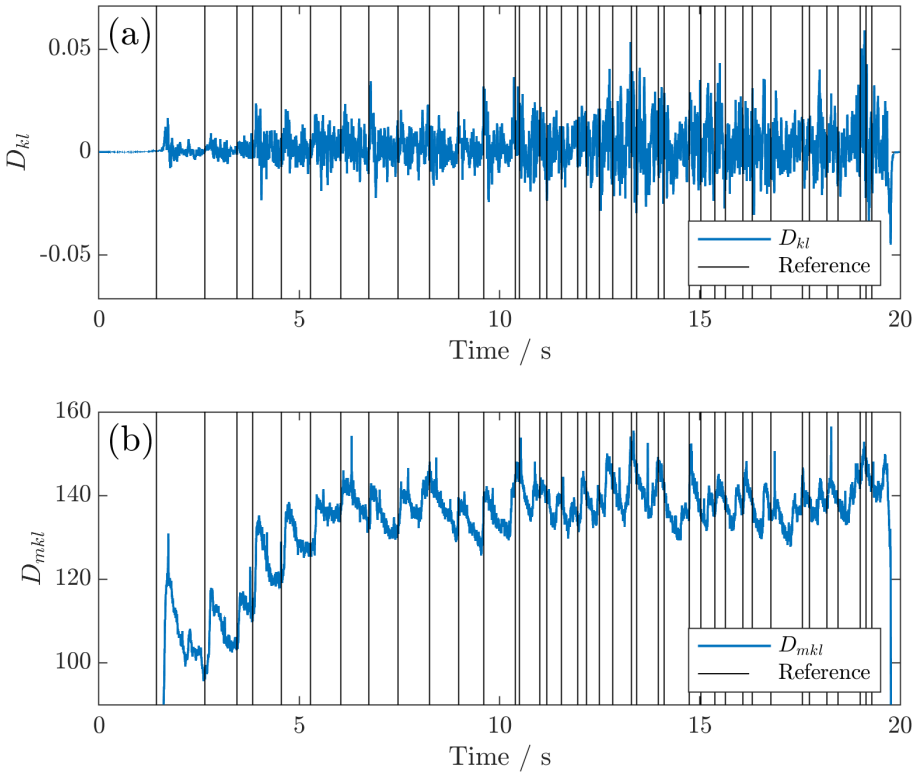


Figure 4.1 – Example and comparison with hand-labeled onset-reference of the
(a) regular
(b) modified Kullback-Leibler divergence (with $\epsilon = 10^{-3}$).

To summarize, the modified Kullback-Leibler divergence is a simple yet suitable measure for onsets. The peaks are sharp and distinctive and no prior knowledge about the signal is necessary. On the downside, different sources can not be distinguished, the harmonic structure of the sound stays unknown and a parameter ϵ has to be chosen by hand.

4.2 Template-based approach

With this rather simple, one-dimensional template-based approach, some drawbacks of the Kullback-Leibler divergence will be addressed, especially to distinguish different components (e.g. different bells) and to gain its harmonic structure. For a successful and robust calculation of the activations, a learning process has to be initiated prior to the online algorithm.

Learning The aim of this learning step is to calculate harmonic templates (e.g. the mean harmonic distribution) of the templates. Ideally, separate recordings of each component exist. At first, the signal is transformed into the time-frequency domain, e.g. by using a CQT (cf. chapter 3) or a STFT. The length/parameters of the transform should also be used in the online calculation afterwards. Therefore, a compromise between accuracy (long transform, so more components can be distinguished eventually) and delay [1] in the online calculation has to be found. Now the transformed signal of the i-th component $|W_i[k,n]|$ can be averaged over time n to obtain a mean magnitude frequency response $\overline{W}_i[k]$, which will be used as the template in the next step. In some cases, a normalization as

$$\overline{W}_i[k] \leftarrow \frac{\overline{W}_i[k]}{\sum_{k=0}^{K-1} \overline{W}_i[k]} \tag{4.3}$$

can be useful, especially when the strength of the activations should be compared. An example for such a one-dimensional template is shown in fig. 4.2.

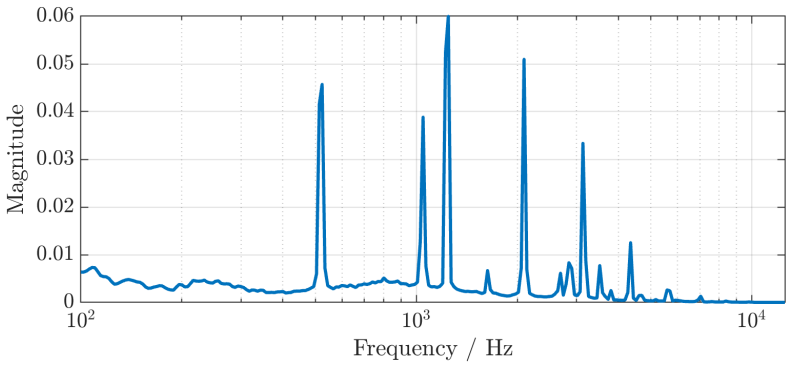


Figure 4.2 – Exemplary one-dimensional template.

Online calculation With the previously obtained templates $\overline{W}_i[k]$, the real-time calculation of the activations $H_i[n]$ is rather simple. The input signal is transformed to

1. The delay is typically determined by the hop size of the frequency transform and not by the transform length itself. However, a small hop size compared to the transform length results again in higher computational complexity.

frequency domain as $|X_i[k,n]|$ using the previously defined parameters. Template and spectrogram are now multiplied element-wise and summed over all K frequency bins as

$$H_i[n] = \sum_{k=0}^{K-1} \overline{W}_i[k] \cdot |X_i[k,n]| \tag{4.4}$$

to obtain the the activations at time n. In matrix-notation, this can be expressed as

$$\mathbf{H}[n] = \begin{bmatrix} H_0[n] \\ H_1[n] \\ \vdots \\ H_{I-1}[n] \end{bmatrix} = diag\left(\overline{\mathbf{W}}^T |\mathbf{X}_I[n]|\right) \tag{4.5}$$

with

$$\mathbf{W} = \begin{bmatrix} \overline{W}_0[0] & \overline{W}_1[0] & \dots & \overline{W}_{I-1}[0] \\ \overline{W}_0[1] & \overline{W}_1[1] & \dots & \overline{W}_{I-1}[1] \\ \vdots & \vdots & \ddots & \vdots \\ \overline{W}_0[K-1] & \overline{W}_1[K-1] & \dots & \overline{W}_{I-1}[K-1] \end{bmatrix}^T \tag{4.6}$$

$$|\mathbf{X}_I[n]| = \begin{bmatrix} |X_0[0,n]| & |X_1[0,n]| & \dots & |X_{I-1}[0,n]| \\ |X_0[1,n]| & |X_1[1,n]| & \dots & |X_{I-1}[1,n]| \\ \vdots & \vdots & \ddots & \vdots \\ |X_0[K-1,n]| & |X_1[K-1,n]| & \dots & |X_{I-1}[K-1,n]| \end{bmatrix}, \tag{4.7}$$

with a total of I components. The activations can also be estimated using a single transformed input signal $|X[k,n]|$ containing signal from all components with

$$\begin{aligned} \mathbf{H}[n] &= \begin{bmatrix} H_0[n] \\ H_1[n] \\ \vdots \\ H_{I-1}[n] \end{bmatrix} = \overline{\mathbf{W}}^T |\mathbf{X}[n]| = \\ &= \begin{bmatrix} \overline{W}_0[0] & \overline{W}_1[0] & \dots & \overline{W}_{I-1}[0] \\ \overline{W}_0[1] & \overline{W}_1[1] & \dots & \overline{W}_{I-1}[1] \\ \vdots & \vdots & \ddots & \vdots \\ \overline{W}_0[K-1] & \overline{W}_1[K-1] & \dots & \overline{W}_{I-1}[K-1] \end{bmatrix}^T \begin{bmatrix} |X[0,n]| \\ |X[1,n]| \\ \vdots \\ |X[K-1,n]| \end{bmatrix}. \end{aligned} \tag{4.8}$$

An example of calculated activations for 4 components can be seen in fig. 4.3.

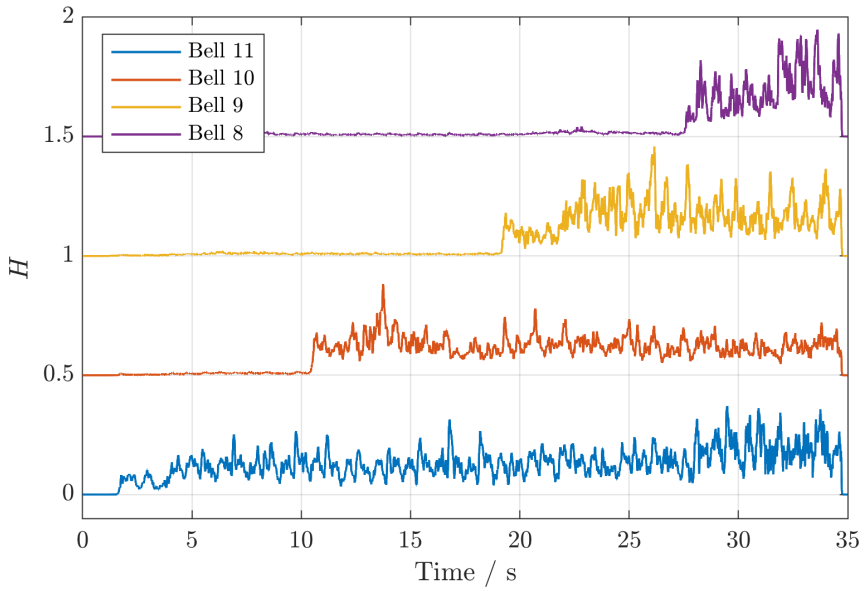


Figure 4.3 – Exemplary calculation of activations using a (sequential) template-based approach with 4 components (with an offset of 0.5 added to every curve for better visibility).

This method can be interpreted as an evaluation of the spectrogram for each component's spectral distribution. E.g. if the template consists of only one non-zero entry at bin k, only the spectrogram's k-th bin weighted with the value of the template contributes to the activations. More non-zero entries result simply in a weighted addition of the spectrogram's spectral components. This approach can be seen as one-dimensional, as the spectrogram is evaluated at only one sample of a frame at a time with a template of only one sample length.

This approach could also be combined with the modified Kullback-Leibler divergence, meaning that the transformed input signal $X[k, n]$ is element-wise weighted with the templates before applying the divergence. However, simulations showed that this leads to even noisier results as well as noticeable influence of crosstalk from other components. An example for this behavior with the same data as in fig. 4.3 is shown in fig. 4.4.

4.2.1 Sequential template-based approach

The above described approach only works, if the isolated sound of each component is available, which is not always the case. However, a specific property of bell-ringing can be used for this implementation. In the case examined, the bells start sequentially [2]. Again, the transformed learning signal (now including all components) is stored in a buffer and shall be denoted as $|X_{\text{buffer}}[k, n]|$.

2. E.g. when using automated ringing systems, the second bell starts ringing 10 s after the first, the third bell starts 10 s after the second and so on.

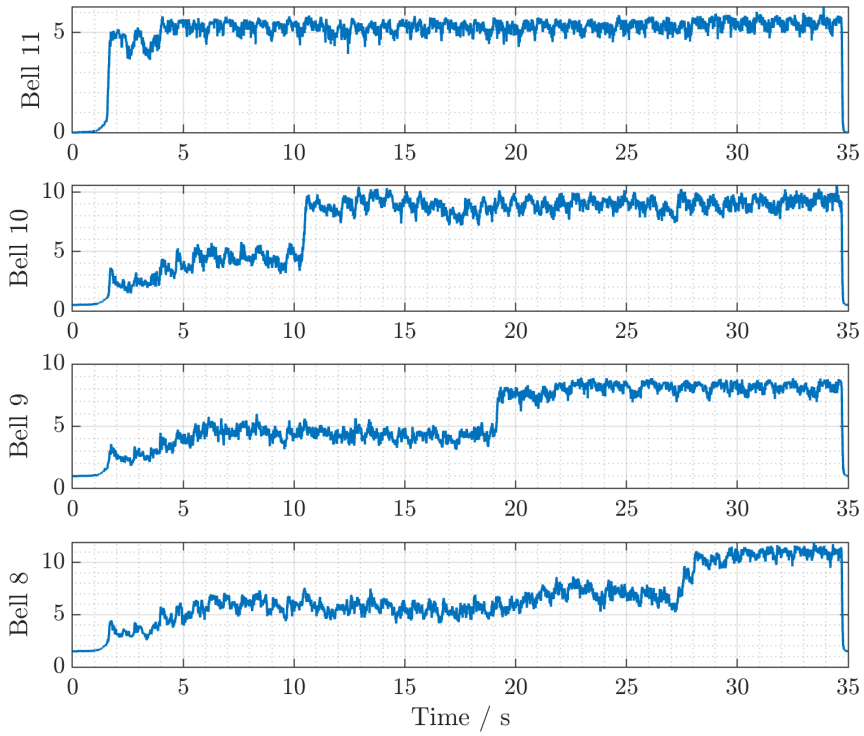


Figure 4.4 – Example of the modified Kullback-Leibler divergence applied to an element-wise weighted spectrum using a sequential template-based approach.

A requirement for this sequential approach is, that the approximate start-time of each component is known. So when observing the period between the start of the first and second component, only the first component is active and the harmonic template $\overline{W}_0[k]$ can be estimated as proposed above. In a next step, only the main components of the template are kept and all spectral components below a certain threshold th are set to zero as

$$\widetilde{W}_i[k] = \begin{cases} \overline{W}_i[k], & \text{if } \overline{W}_i[k] > th * \max\left\{\overline{W}_i\right\}, \\ 0, & \text{else,} \end{cases} \tag{4.9}$$

and afterwards normalized. In different test scenarios, a threshold in the range of $th \in [0.1, 0.2]$ was sufficient.

After this "gated" template of the first component is estimated, we can delete its dominant frequency components for all analyzed time instances n in the spectrogram $|X_{\text{buffer}}[k,n]|$. This can be done as

$$|X_{\text{buffer}}[k,n]| \leftarrow \begin{cases} |X_{\text{buffer}}[k,n]|, & \text{if } \widetilde{W}_i[k] = 0, \\ 0, & \text{else.} \end{cases} \quad \forall k,n \tag{4.10}$$

An exemplary template is shown in fig. 4.5.

Now we can continue with the second time step (between the start of the second and the third component). In this time region, (approximately) only the second component should be active. The above written steps can now be repeated to find all required templates

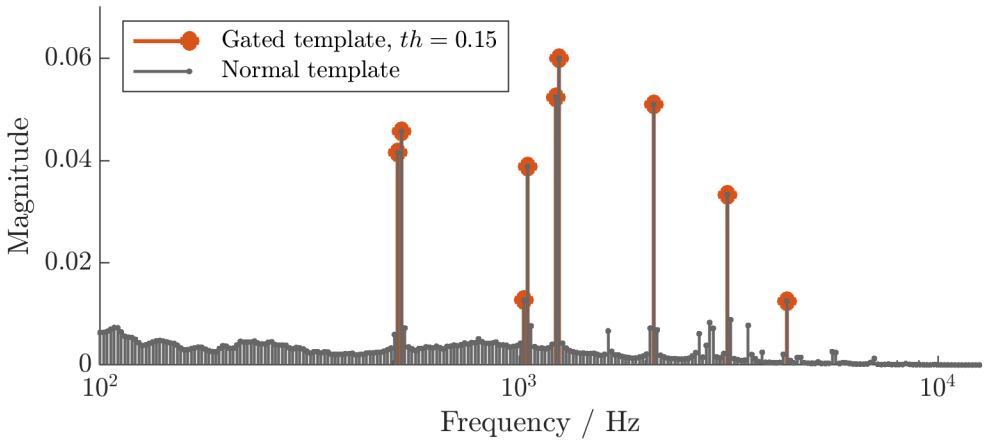


Figure 4.5 – Comparison of ungated and gated templates for the sequential template-based approach (not normalized).

sequentially. The whole process is shown in fig. 4.6.

The rather simple template-based approach for distinguishing components and calculating its activation works pretty well for a limited number of components and an appropriately set threshold. It has its limits, as no cost function or sparsity constraint is invoked. This results in noticeable influence of crosstalk between components, especially with an increasing number of components to be identified. The sequential algorithm works only under the presupposition of sequential starts and moderate overlapping in the frequency domain of components. However, in test-cases with bell ringing in a real-life scenario [3], up to 5 components could be reliably distinguished.

3. Recording of Graz Dom and Mausoläum with wind noises, provided by ORF Landesstudio Steiermark

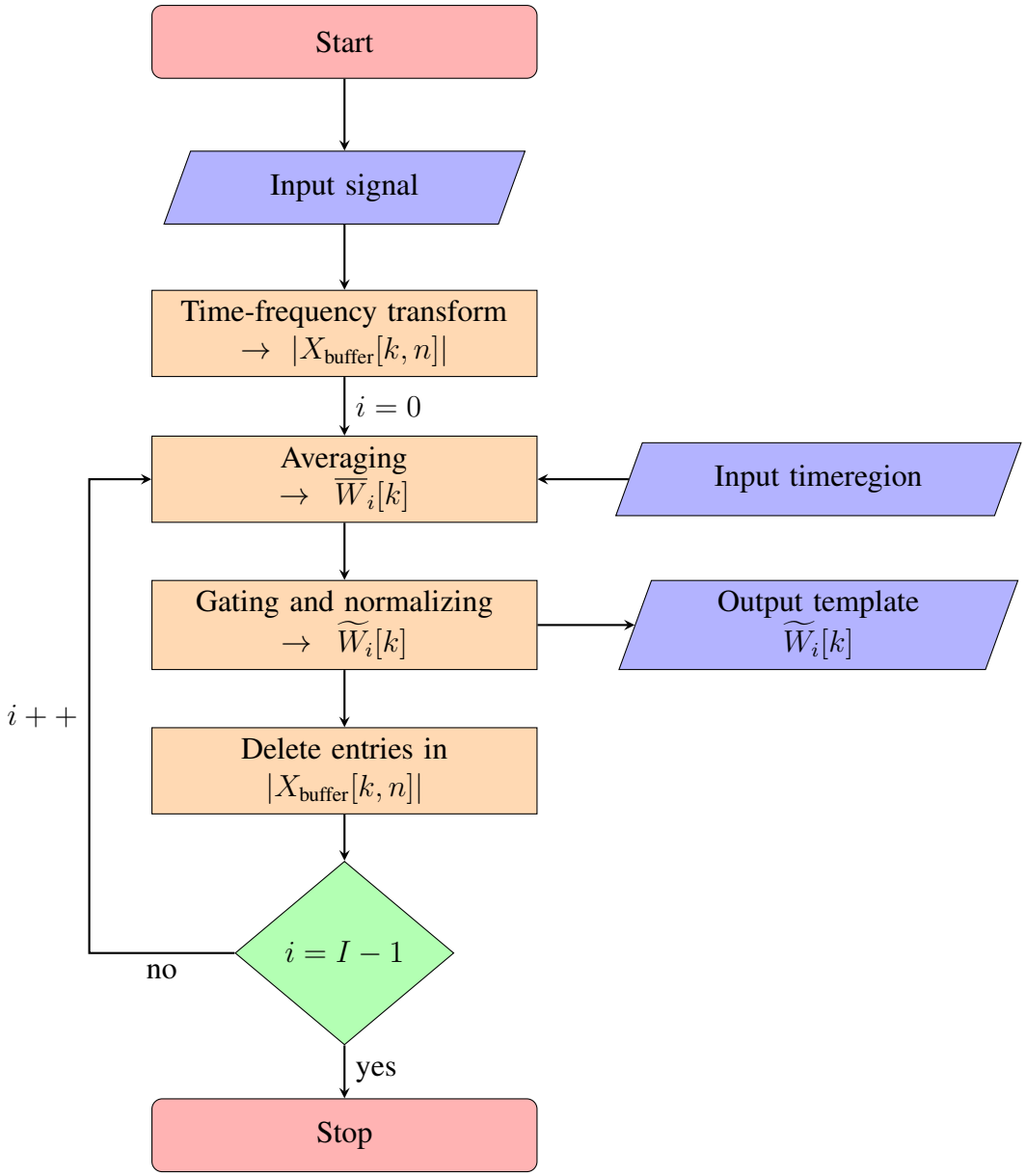


Figure 4.6 – Flowchart for the learning process of the sequential template-based approach.

4.3 Non-negative Matrix Factorization (NMF)

4.3.1 Standard NMF

Non-negative matrix factorization (NMF) or non-negative matrix approximation is a multivariate analysis technique for blind source separation and widely used in the fields of computer vision and audio signal processing. The method found wide dissemination in 1999 after Lee and Seung investigated its properties and found more convenient algorithms [LS99]. As more and more variants and optimizations for this algorithm turned up in the last several years, only basic assumptions and approaches used in this application are presented. The interested reader may be referred to the comprehensive book by

Cichocki et.al. [Cic+09].

The basic idea is to decompose a given input matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{F \times N}$, e.g. a magnitude spectrogram of an audio signal obtained by a CQT, into a product of two positive matrices $\mathbf{W} \in \mathbb{R}_{\geq 0}^{F \times C}$ (dictionary or spectral templates) and $\mathbf{H} \in \mathbb{R}_{\geq 0}^{C \times N}$ (temporal activations) such that

$$\mathbf{X} \approx \mathbf{WH} = \mathbf{U}\,, \tag{4.11}$$

where $F \in \mathbb{N}$ is the feature dimensionality, $N \in \mathbb{N}$ the number of observations and $C \in \mathbb{N}$ the number of components. Typically the rank C needs to be determined beforehand and is with $C < FN/(F+N)$ substantially smaller than F and N, therefore we speak of a low-rank approximation [LS99]. In a musical context using a magnitude spectrogram as input matrix, the dictionary $\mathbf{W}$ can be seen as the harmonic structure, whereas $\mathbf{H}$ represents the activations over time of each C reoccurring components. The components can e.g. be different notes/chords and/or parts played by different instruments. Due to the non-negativity constraint, only additive combinations are possible, whereas subtractive combinations cannot occur. This behavior matches also with our intuitive understanding of music as sum of several sound events.

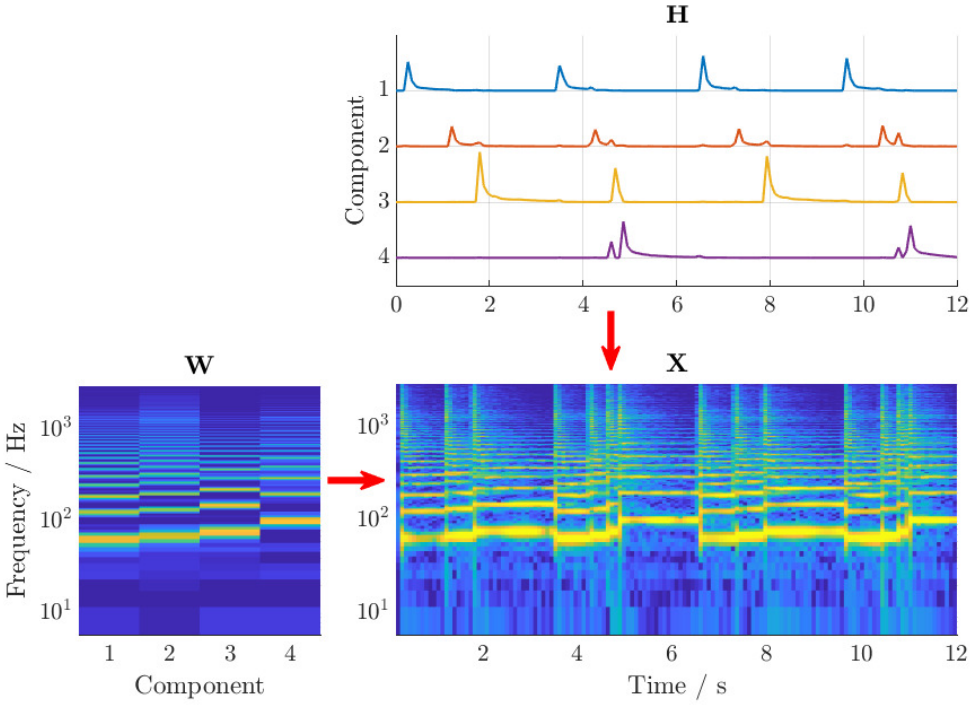


Figure 4.7 – NMF using an STFT of length 1024, 50% overlap, Itakura-Saito distance and 100 iterations of the intro of "Birdland" by Weather Report, downsampled to $f_s = 6\,\mathrm{kHz}$.

As this yields in an optimization problem, a cost function

$$\min\{D(\mathbf{X}, \mathbf{U})\} = \min_{\mathbf{W},\mathbf{H} \geq 0}\{D(\mathbf{X}, \mathbf{WH})\} \tag{4.12}$$

can be defined and minimized iteratively to find an optimal solution for $\mathbf{W}$ and $\mathbf{H}$ under the constraint that both matrices consist of non-negative elements. In the following, several cost functions are presented as well as a multiplicative, iterative solution without the need of a step-size parameter.

Cost functions

The choice of a suitable cost function is crucial for a successful approximation. The most utilized function for NMF is the Kullback-Leibler divergence, a majority of the original algorithms are based on it. Also Euclidean distance and Itakura-Saito divergence are extensively used recently. All those mentioned functions belong to the group of Bregman divergences and can be combined to the beta-divergence, which enables us to blend these continuously [Cic+09]. The following divergences are described in a universal manner using observations $\mathbf{p}$, its estimates $\mathbf{q}$ (which correspond to $\mathbf{U}$ in the NMF) and an index i. Note that only a short selection is presented, a comprehensive compilation can be found in [Cic+09, Chapter 2].

Squared Euclidean distance The squared Euclidean distance, also known as ℓ_2-norm, is merely the sum of the quadratic estimation errors and defined as

$$D_2(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2^2 = \sum_i (p_i - q_i)^2. \tag{4.13}$$

Due to its simple, intuitive calculation and its optimality for Gaussian error signals it is extensively used for low-dimensional data. However, as outliers influence the result drastically and it is not suitable for high-dimensional data, the usage must be handled with care [Cic+09, p. 83].

Generalized Kullback-Leibler divergence Initially, the Kullback-Leibler divergence was a measure for the (dis-)similarity of two probability distributions. It turned out as a suitable measure in many machine learning applications like the NMF. One form of the Kullback-Leibler divergence for spectral changes over time has already be addressed in section 4.1. The main difference of the extended (generalized) KL-divergence

$$D_{\mathrm{gKL}}(\mathbf{p}, \mathbf{q}) = \sum_i \left(p_i \cdot \ln\left(\frac{p_i}{q_i}\right) - p_i + q_i \right) \tag{4.14}$$

to the previously discussed version is the supplementary subtraction of the estimation error.

Itakura-Saito distance The Itakura-Saito distance

$$D_{\text{IS}}(\mathbf{p}, \mathbf{q}) = \sum_i \left(\ln\left(\frac{q_i}{p_i}\right) + \frac{p_i}{q_i} - 1 \right) \tag{4.15}$$

was originally used as a measure for similarity of a maximum likelihood estimation for short-time speech spectra. Due to its good resemblance of perceptual properties it has become a standard measure in speech signal processing [Cic+09, p. 113].

Beta-divergence The beta-divergence can be seen as an attempt to connect the squared Euclidean distance with the Itakure-Saito distance smoothly via a parameter β. On the transition, the Kullback-Leibler divergence is reached. After it was introduced by Eguchi et.al. [EK01], it could be utilized successfully for the NMF [Kom07].

The measure is defined as

$$D_{\text{B}}^{(\beta)}(\mathbf{p}, \mathbf{q}) = \begin{cases} \sum_i \left(\ln\left(\frac{q_i}{p_i}\right) + \frac{p_i}{q_i} - 1 \right), & \text{if } \beta = -1, \\ \sum_i \left(p_i \cdot \ln\left(\frac{p_i}{q_i}\right) - p_i + q_i \right), & \text{if } \beta = 0, \\ \sum_i \left(p_i \frac{p_i^\beta - q_i^\beta}{\beta} - \frac{p_i^{\beta+1} - q_i^{\beta+1}}{\beta+1} \right), & \text{else} \end{cases} \tag{4.16}$$

with $\beta \in \mathbb{R}$, whereas the subscript B indicates the underlying beta-divergence. Special cases are listed in table 4.1 as well as a graphical representation in fig. 4.8. The interested reader shall find its derivation in [Cic+09, Section 2.6][EK01].

β	Divergence
1	Squared Euclidean distance
0	Kullback-Leibler divergence
-1	Itakura-Saito distance

Table 4.1 – Special cases of the beta-divergence.

Update equations

As a cost function is defined, typically an iterative gradient descent search for the optimal solution in a form

$$\begin{aligned} \mathbf{W} &\leftarrow \mathbf{W} - \mu \nabla_{\mathbf{W}} D(\mathbf{X}, \mathbf{WH}) \\ \mathbf{H} &\leftarrow \mathbf{H} - \mu \nabla_{\mathbf{H}} D(\mathbf{X}, \mathbf{WH}) \end{aligned} \tag{4.17}$$

with a step-size μ and the gradient of the cost function ∇D is used. As the ability to converge and the convergence time depends on an appropriate choice of μ with respect

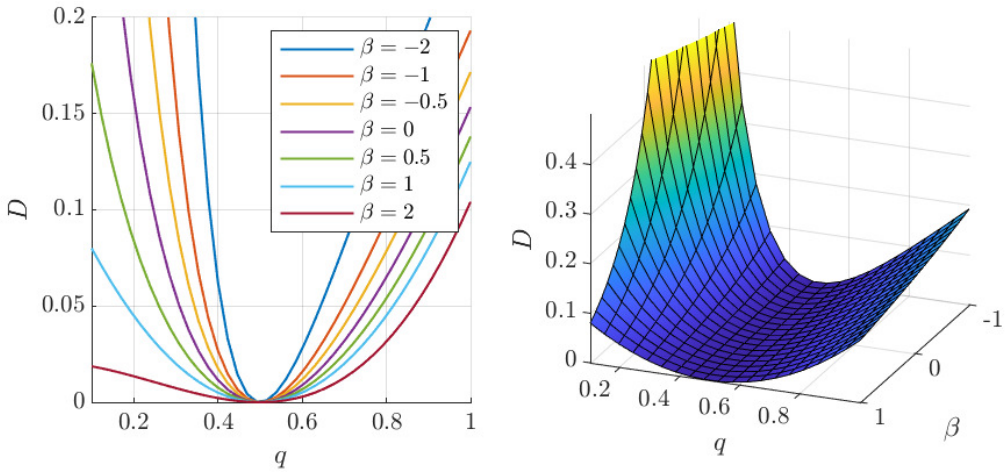


Figure 4.8 – Examples for the beta-divergence with $p = 0.5$ and variable β.

to the input data, it is not very practical. Therefore, Lee and Seung [LS01] introduced multiplicative update rules based on the KL-divergence without the need of a step-size μ as

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \left((\mathbf{X} \oslash \mathbf{WH})\, \mathbf{H}^T \right) \oslash \left(\mathbf{1}_{F \times N} \mathbf{H}^T \right) \tag{4.18}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \left(\mathbf{W}^T (\mathbf{X} \oslash \mathbf{WH}) \right) \oslash \left(\mathbf{W}^T \mathbf{1}_{F \times N} \right) \tag{4.19}$$

with the Hadamard product $\otimes$, the Hadamard division $\oslash$ and $\mathbf{1}$ as a matrix of ones with $\dim(\mathbf{1}) = \dim(\mathbf{X})$. Usually $\mathbf{W}$ and $\mathbf{H}$ are normalized after updating. Note, that this multiplicative update must not necessarily have the same convergence properties as an additive approach and it is not a strictly convex problem anymore, as e.g. values set to 0 cannot be modified any further.

The update algorithm in eqs. (4.18) and (4.19) is only valid for KL-divergence. However, the update equations can be extended to a universal form for the beta-divergence [FBD09; DCL10] as

$$\mathbf{W} \leftarrow \mathbf{W} \otimes \left(\left(\mathbf{X} \otimes (\mathbf{WH})^{\circ(\beta-1)} \right) \mathbf{H}^T \right) \oslash \left((\mathbf{WH})^{\circ(\beta)}\, \mathbf{H}^T \right) \tag{4.20}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \left(\mathbf{W}^T \left(\mathbf{X} \otimes (\mathbf{WH})^{\circ(\beta-1)} \right) \right) \oslash \left(\mathbf{W}^T (\mathbf{WH})^{\circ(\beta)} \right) \tag{4.21}$$

with the Hadamard power $(\cdot)^{\circ(\cdot)}$ and $\beta \in \mathbb{R}$. With the choice of β, an emphasis can be put on spectral components with high resp. low energy. Only the Itakura-Saito divergence is scale-invariant in the class of beta-divergences, resulting in the same penalization of too small or too large coefficients. With $\beta < -1$ more emphasis is put on spectral components with low energy, whereas with $\beta > -1$ the opposite effect can be observed [DCL10].

Initialization

As $\mathbf{W}$ and $\mathbf{H}$ are updated in a multiplicative sense in each iteration (cf. eqs. (4.20) and (4.21)), the initial values have to be chosen with care as only non-zero elements can be changed. It can not even be anticipated that the objective function is strictly convex in a multivariate environment, local minima can occur. Of course also convergence time depends strongly on the initial values [Cic+09, Section 1.3.3].

Two rather simple approaches are to initialize $\mathbf{W}$ and $\mathbf{H}$ with positive (pseudo-)random numbers or unitarily. A way to ensure "robust" initial values is to initialize in fact several matrices and run a NMF with only few iterations (and maybe even with a smaller dataset to accelerate this process). The combination of $\mathbf{W}$ and $\mathbf{H}$ with the smallest cost is chosen as starting point for the actual NMF [Cic+09, Section 1.3.3].

A different, promising approach is to utilize prior knowledge of the signal and the components [Dri+13]. So an emphasis with higher initial coefficients can be put on approximate locations in $\mathbf{H}$ where a special component is active. This can be done vice versa with the templates $\mathbf{W}$ in the spectral domain, e.g. by setting coefficients out of the components range to 0 and using an estimated spectral distribution as initial value.

4.3.2 Non-negative Matrix Factor Deconvolution (NMFD)

The algorithm discussed in section 4.3.1 is only valid for "one-dimensional" templates, therefore, each component represents one column in $\mathbf{W}$ and is only one sample long. This approach can work for a majority of melody instruments where only minor temporal variation occur. Especially for percussion instruments and sounds with strongly different decay times of its partial tones, no sufficient results can be anticipated. Therefore, Smaragdis [Sma04] proposed to extend the NMF model in eq. (4.11) to

$$\mathbf{X} \approx \sum_{t=0}^{T-1} \mathbf{W}_t \overset{t\rightarrow}{\mathbf{H}} = \mathbf{U} \tag{4.22}$$

where $\mathbf{X}$ and $\mathbf{H}$ have the same dimensions as before, but with $\mathbf{W} \in \mathbb{R}_{\geq 0}^{F \times C \times T}$ as a tensor of order 3. The additional dimension takes a temporal course of each component with a length of T samples into account. An example can be seen in fig. 4.9. The notation in eq. (4.22) indicates $\mathbf{W}_t = \mathbf{W}[:, :, t]$, therefore we use the t-th "slice" of the tensor. $\overset{t\rightarrow}{(\cdot)}$ denotes a frame-shift operator, where the columns of a matrix are shifted by t spots to the right. All vacant positions are filled with zeros to maintain its dimensions. In the same manner can a shift to the left $\overset{\leftarrow t}{(\cdot)}$ be described. An example is given in eq. (4.23).

$$\mathbf{A} = \overset{0\rightarrow}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \overset{1\rightarrow}{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \end{bmatrix}, \quad \overset{2\rightarrow}{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 4 \end{bmatrix}, \quad \overset{\leftarrow 1}{\mathbf{A}} = \begin{bmatrix} 2 & 3 & 0 \\ 5 & 6 & 0 \end{bmatrix} \tag{4.23}$$

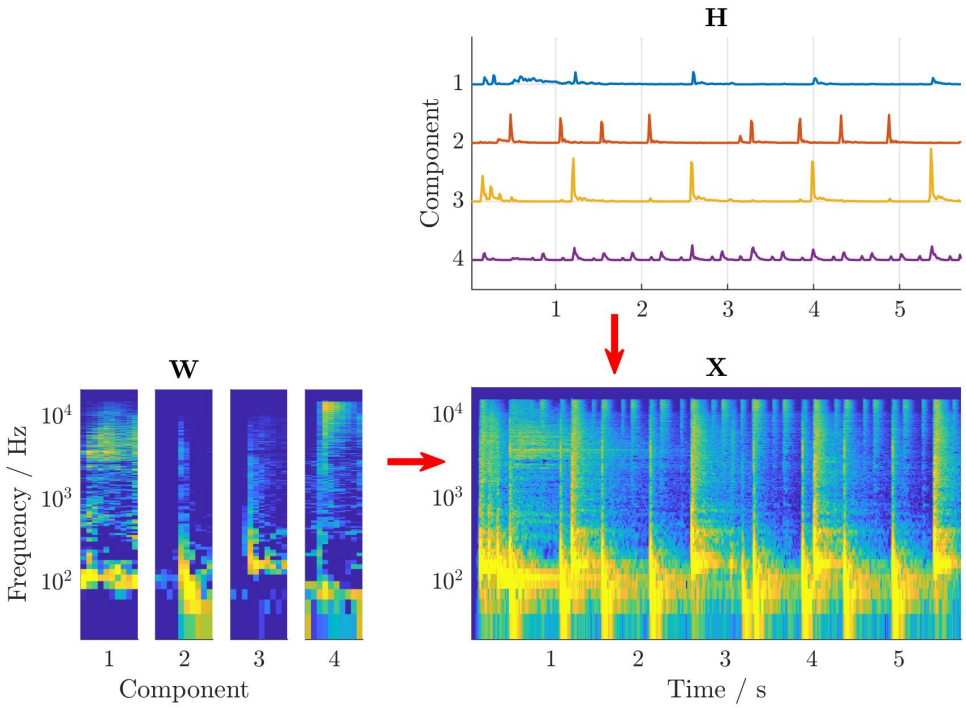


Figure 4.9 – Convolutive NMF using an STFT of length 2048, 75% overlap, $\beta = 0$ and 100 iterations of the drum-intro of "Rosanna" by Toto. The components can be interpreted as ① Ride, ② Bass drum, ③ Snare drum and ④ closed Hi-Hat.

Update equations

The NMFD can again be updated using a multiplicative approach. The initially introduced algorithm by Smaragdis [Sma04] is based on the generalized Kullback-Leibler divergence and defined as

$$\mathbf{W}_t \leftarrow \mathbf{W}_t \otimes \left((\mathbf{X} \oslash \mathbf{U}) \overset{t\rightarrow T}{\mathbf{H}} \right) \oslash \left(\mathbf{1}_{F \times N} \overset{t\rightarrow T}{\mathbf{H}} \right), \qquad \forall t \in [0, \ldots, T-1], \tag{4.24}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \overset{\longleftarrow t}{(\mathbf{X} \oslash \mathbf{U})} \right) \oslash \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \mathbf{1}_{F \times N} \right), \quad \forall t \in [0, \ldots, T-1], \tag{4.25}$$

with the definition of $\mathbf{U}$ from eq. (4.22). One can see its resemblance to the standard NMF in eqs. (4.18) and (4.19). When using the NMFD, not only two matrices will be optimized, but a set of $T + 1$ matrices [4], resulting in higher computational complexity.

4. $\mathbf{H}$ and T slices of $\mathbf{W}$

In the same manner as with standard NMF, Cichocki adapted the algorithm in eqs. (4.18) and (4.19) for a generalized beta-divergence [Cic+09, Section 3.2], calling it "Convolutive NMF". The update rules are given as

$$\mathbf{W}_t \leftarrow \mathbf{W}_t \otimes \left(\left(\mathbf{X} \otimes \mathbf{U}^{\circ(\beta-1)} \right) \overset{t\rightarrow}{\mathbf{H}}^T \right) \oslash \left(\mathbf{U}^{\circ(\beta)} \overset{t\rightarrow}{\mathbf{H}}^T \right), \quad \forall t \in [0, \ldots, T-1], \tag{4.26}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \overset{\longleftarrow t}{\left(\mathbf{X} \otimes \mathbf{U}^{\circ(\beta-1)} \right)} \right) \oslash \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \overset{\longleftarrow t}{\mathbf{U}^{\circ(\beta)}} \right). \tag{4.27}$$

Additional constraints

It can be quite practical in various cases to define additional constraints for the cost function. Two major objectives are usually sparseness of $\mathbf{H}$ as well as orthogonality of $\mathbf{W}$. Generally speaking, the cost function in eq. (4.16) can be extended to

$$D_{\mathrm{Bc}}^{\beta}(\mathbf{X}, \mathbf{WH}) = D_{\mathrm{B}}^{\beta}(\mathbf{X}, \mathbf{WH}) + \lambda_{\mathbf{H}} J_{\mathbf{H}}^{sp} + \lambda_{\mathbf{W}} J_{\mathbf{W}}^{o} \tag{4.28}$$

with additional penalty terms for special properties of $\mathbf{H}$ and $\mathbf{W}$, each scaled with a user-defined scalar parameter λ.

The sparsity constraint attempts to "enforce" sparsity for the rows of $\mathbf{H}$. This can be quite easily done by calculating the ℓ_1-norm of $\mathbf{H}$, which is in this non-negative case merely the sum of all its entries

$$J_{\mathbf{H}}^{sp} = \sum_{c=0}^{K-1} \sum_{n=0}^{N-1} \mathbf{H}[k, n]. \tag{4.29}$$

Note that valid results can only be expected if $\mathbf{W}$ is normalized, otherwise this constraint would lead to $\mathbf{H} \to \mathbf{0}$ and $\mathbf{W} \to \infty$ as it aims to minimize $\mathbf{H}$.

The orthogonality constraint can be used to reduce correlation between the columns (and therefore components) of $\mathbf{W}$. The penalty term can now be expressed as

$$J_{\mathbf{W}}^{o} = \sum_{f=0}^{F-1} \sum_{c=0}^{K-1} \frac{1}{2} \sum_{p \neq q} \mathbf{W}_p^T \mathbf{W}_q. \tag{4.30}$$

Cichocki et.al. [Cic+09, Section 3.7.2] included the two objectives defined in eqs. (4.29) and (4.30) into the update algorithm eqs. (4.26) and (4.27), such that a general from can be expressed as

$$\mathbf{W}_t \leftarrow \mathbf{W}_t \otimes \left(\left(\mathbf{X} \otimes \mathbf{U}^{\circ(\beta-1)} \right) \overset{t\rightarrow}{\mathbf{H}}^T \right) \oslash \left(\mathbf{U}^{\circ(\beta)} \overset{t\rightarrow}{\mathbf{H}}^T + \lambda_{\mathbf{W}} \sum_{q \neq t} \mathbf{W}_q \right), \tag{4.31}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \overset{\longleftarrow t}{\left(\mathbf{X} \otimes \mathbf{U}^{\circ(\beta-1)} \right)} \right) \oslash \left(\sum_{t=0}^{T-1} \mathbf{W}_t^T \overset{\longleftarrow t}{\mathbf{U}^{\circ(\beta)}} + \lambda_{\mathbf{H}} \mathbf{1}_{K \times N} \right). \tag{4.32}$$

4.3.3 Efficient real-time calculation

It is obvious, that the NMFD is a computationally quite complex process, which works in the presented form only offline. Although approaches for online calculation exist, e.g. in [WR17], this computational expensive process may not be necessary in this specific case. As we are only interested in the activations in a real-time context, we can utilize a prior calculated and static dictionary $\mathbf{W}$.

Let us look at this problem in a component-wise and block-based view. The NMFD-template of the c-th component with length T and F features shall be defined as $\mathbf{W}^{(c)} = \mathbf{W}[:, c, :]$ and $\mathbf{W}^{(c)} \in \mathbb{R}_{\geq 0}^{F \times T}$.

The activations $\widetilde{\mathbf{H}}_c$ of the c-th element can now be estimated quite easily for each sample n as

$$\widetilde{\mathbf{H}}_c[n] = \sum_{t=0}^{T-1} \sum_{f=0}^{F-1} \mathbf{W}_t^{(c)} \otimes \mathbf{X}[n+t], \tag{4.33}$$

whereas an additional delay of $T-1$ samples must be introduced for a real-time calculation to ensure causality.

Usually the activations as obtained in eq. (4.33) will vary from those of eq. (4.22), as the latter is based on an approximation, not a "direct" calculation. This can be easily understood when rewriting eq. (4.22) as

$$\sum_{t=0}^{T-1} \mathbf{W}_t \overset{t\rightarrow}{\mathbf{H}} = \mathbf{U} \approx \mathbf{X} = \sum_{t=0}^{T-1} \mathbf{W}_t \overset{t\rightarrow}{\mathbf{H}} + \mathbf{E} \tag{4.34}$$

with an error $\mathbf{E}$ and $\dim(\mathbf{E}) = \dim(\mathbf{X})$. It is clear, that the method in eq. (4.33) is influenced by $\mathbf{E}$ resulting in noisier or smeared activations. An example and comparison for this behavior can be seen in fig. 4.10. The peaks obtained by the online calculations are in this example not as sharp and defined than those of the offline calculation.

4.4 Onset-detection

When examining activations, e.g. obtained by algorithms in sections 4.1 to 4.3, the automated detection of onsets can be of interest. All now discussed methods are based on the idea to define a threshold. If a sample lies over this threshold, an onset is detected. As

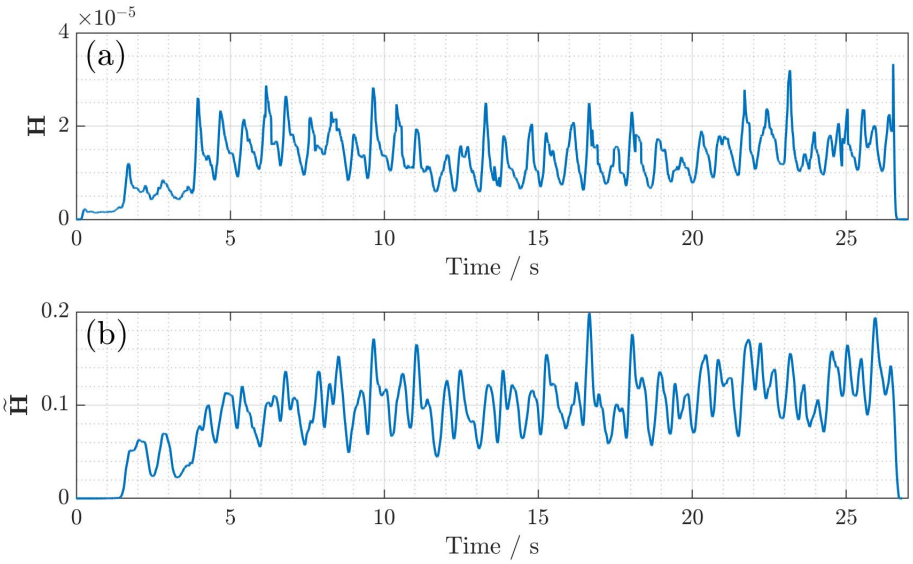


Figure 4.10 – Comparison for (a) offline and (b) online calculation of activations for one bell of Graz Mausoläum.

simple approaches with a static detection threshold have many drawbacks, such as strong influence of the signal levels, dynamic peak-picking algorithms are preferred.

Dynamic algorithms aim to set an individual threshold for each sample, depending on neighboring values. In a rather simple yet effective moving-median approach, the threshold th at position n is defined as the median of a arbitrary cost function D in a rectangular window w_n, symmetrically distributed around the sample to be examined [BS03; Dux+03; Kau02]. This can be expressed as

$$th[n] = C_{th} \cdot \text{median}(D(w_n)), \qquad w_n \in \left[n - \frac{L-1}{2}; n + \frac{L-1}{2}\right] \tag{4.35}$$

with a window-length L and a predefined, constant weighting-factor C_{th}. A low value of C_{th} increases the number of (false) detections, while a higher value leads to a more strict selection [BS03]. Note that the window w_n must contain an odd number of samples in order to be symmetric around sample n.

The constant weighting-factor C_{th} in eq. (4.35) still leads to a not neglectable influence of the signal level for detected onsets. Especially at low amplitudes, this results in a number of false detections. The optimal value of the weighing factor can be modeled using a sigmoid-function [Kli04, Section 4.2.3] and calculated sample-wise as

$$C_{th}[n] = \frac{1}{1 + e^{\text{median}(D(w_n))}} + u \tag{4.36}$$

with a predefined offset u. The threshold can now be calculated as

$$th_s[n] = C_{th}[n] \cdot \text{median}(D(w_n)) = \left(\frac{1}{1 + e^{\text{median}(D(w_n))}} + u\right) \cdot \text{median}(D(w_n)). \tag{4.37}$$

An example for this approach is shown in fig. 4.11.

It should be additionally mentioned, that instead of a moving-median also a moving-average approach is suitable. So in eqs. (4.35) to (4.37) the mean value is used instead of the median.

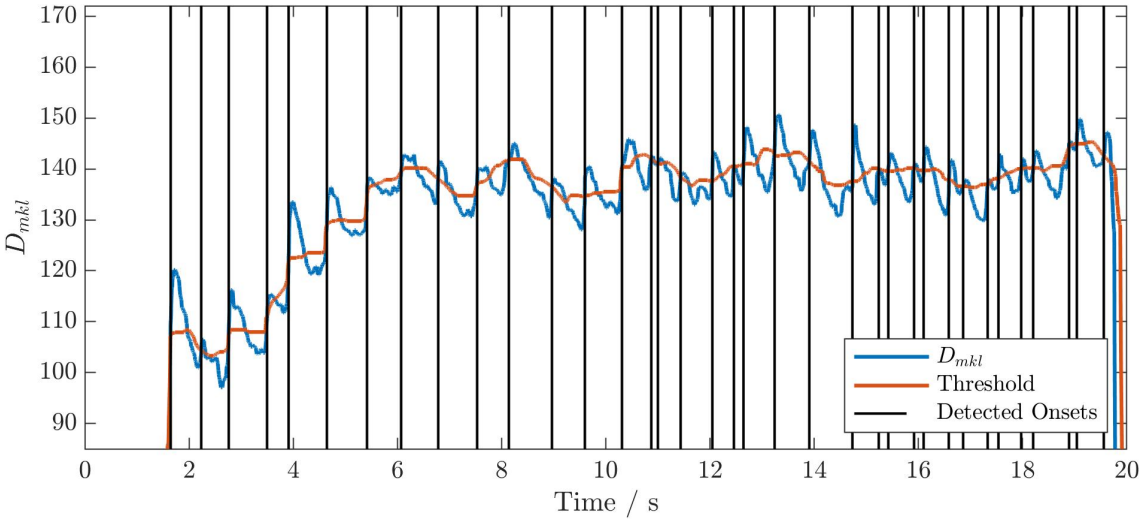


Figure 4.11 – Onset-detection as defined in eq. (4.37) with the cost function in fig. 4.1(b) with $u = 1.01$ and $L = 501$.

Chapter 5

Consonance

To add another layer of information to the signal analysis, the harmonic structure of the signal can be taken into account. A convenient and for normal hearing people often subconsciously perceived feature is the consonance (respectively dissonance) of a signal.

As psychoacoustical parameter, even an exact definition of consonance can be difficult. It is often referred to as “pleasantness” or “acceptability”. The categorization and degree of consonance is also highly influenced by cultural background and musical education [LE20].

Despite all difficulties, some approaches for an objective calculation of this parameter exist. In the following, the method of local consonance by Sethares [Set93] will be explained and expanded to a STFT-based, frequency selective algorithm.

5.1 Historical models and explanations of consonance and dissonance

The definition and cause of consonance and dissonance in music has been a topic of interest for scientist for centuries. First systematic studies were conducted in, the 16th century by renowned scientists like Galilei, Leibniz or Euler, finding that some fixed frequency ratios between two tones are perceived more consonant than others [PL65]. A quite early yet surprisingly accurate theory regarding dissonance and consonance has been defined by Helmholtz [Hel63]. He stated, that beating between two pure sine tones can be heard for a small frequency offset. If that difference increases, rapid beating or roughness is perceived with its maximum at about $30\,\mathrm{Hz}$ to $40\,\mathrm{Hz}$ difference [1], and with even more frequency difference two separate tones are heared. It is stated, among other things, that

1. It was found later, that the frequency difference for maximal roughness is frequency dependent [PL65; Sot94].

dissonance between two or more tones is related to the beats and roughness of adjacent partials. He was also able to derive fixed frequency ratios for consonant and dissonant intervals using this theory, which correspond well with our common perception of musical intervals [Hel63, chapter 10]. As rule of thumb, smaller numbers in the frequency ratio result in a more consonant sound that higher numbers, e.g. a ratio of 1:2 (octave) sounds more pleasant than 2:3 (pure fifth), or even 5:9 (major seventh) [PL65]. It was found in later studies, that not only the partials have to be taken into account, but also the resulting difference tones [Kru03].

Plomp and Levelt [PL65] studied and tested this hypothesis regarding *local consonance* [2] thoroughly and found, that the perception of roughness (and according to Helmholtz therefore dissonance) is actually related to the critical bandwidth of the human ear. The perception can generally be described with the highest consonance being perceived at unison, a relatively small frequency offset results in highest dissonance, which resolves again in consonant sound with increasing frequency difference. The maximal dissonance between two tones can be found at a difference of $25\,\%$ of the critical bandwidth, whereas maximal consonance can be modeled at above $100\,\%$ of the critical bandwidth [PL65]. The resulting model for consonance and dissonance of pure tones can be seen in fig. 5.1.

Sethares [Set93] used the latter theory by Plomp and Levelt [PL65] to derive a simple and parametrized mathematical model for consonance.

5.2 Local consonance model by Sethares

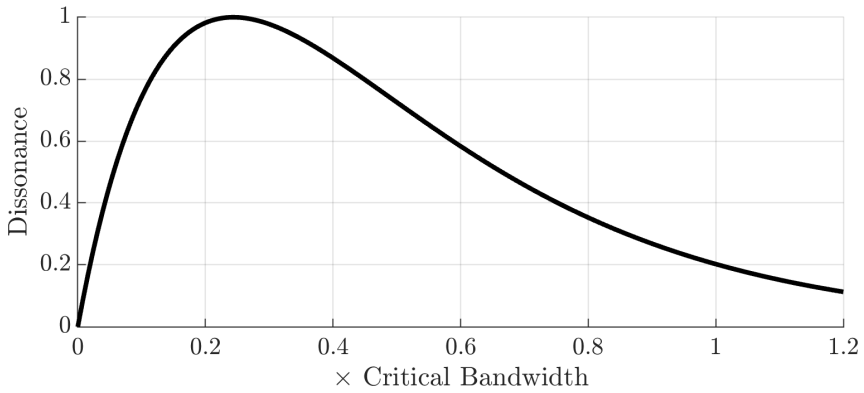
Initially, the intention of Sethares [Set93] was to find optimal, possibly non-harmonic timbres for arbitrary scales. To do this, the model of local consonance by Plomp and Levelt [PL65] was parametrized and a relatively simple calculation was derived to calculate a measure for dissonance.

Parametrization The dissonance model of [PL65] can be parametrized in the form

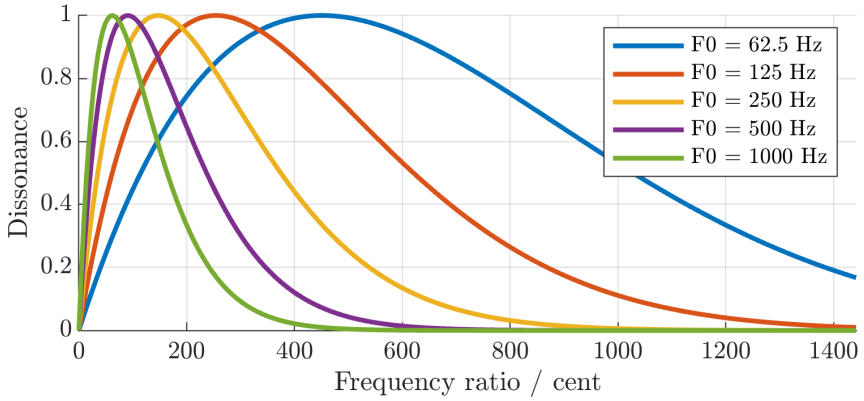
$$d(x) = e^{-ax} - e^{-bx} \tag{5.1}$$

where $d(x)$ describes an unscaled dissonance function and x the frequency difference between two sine tones. Both exponential factors can be found when averaging the model in fig. 5.1 for different frequencies, resulting in

2. When speaking of local consonance, only isolated tone pairs are examined without musical context.



(a) Dissonance dependent on the critical bandwidth according to [PL65].



(b) Dissonance dependent on the fundamental frequency $F0$ [Set93].

Figure 5.1 – Modeled dissonance between two pure tones.

$$a = 3.5 \text{ and } b = 5.75\,. \tag{5.2}$$

The parametrization can be further extended to work conveniently with the two frequencies f_1 and f_2 as well as their amplitudes v_1 and v_2. To do so, it must be ensured that $f_1 < f_2$, and the point of maximal dissonance $d^* = 0.24$ according to [PL65; Set93] is introduced. Hence, an extended parametrization

$$d(f_1, f_2, v_1, v_2) = v_{12}\left(e^{-as(f_2-f_1)} - e^{-bs(f_2-f_1)}\right), \quad \text{with } s = \frac{d^*}{s_1 f_1 + s_2} \tag{5.3}$$

can be formulated. The combined amplitude coefficient

$$v_{12} = v_1 v_2 \tag{5.4}$$

ensures, that tones with lower amplitude contribute quantitatively less to the dissonance. The additional parameters s_1 and s_2 can be used to enforce a frequency dependent calculation based on an approximated model of the critical bandwidth as intended by Plomp [PL65]. Hence, for lower frequencies the dissonance curve depending on the absolute frequency difference is compressed, whereas for higher frequencies it is stretched. This

is done to ensure, that maximal dissonance occurs at $25\,\%$ of the critical bandwidth. In a least-squares search, the values [Set93]

$$s_1 = 0.021 \;\text{ and }\; s_2 = 19 \tag{5.5}$$

were found.

Now when assuming, that dissonance between complex tones has a cumulative property [Set93], an overall dissonance D_F can be calculated. Let us assume a timbre F with N partials with frequencies $f_0 < f_1 < \cdots < f_{N-1}$. The total dissonance is now the sum of the local dissonance between each individual partial, namely

$$\begin{aligned} D_F &= \frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} d(f_i, f_j, v_i, v_j) \\ &= \frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} v_{ij}\left(e^{-as|f_j-f_i|} - e^{-bs|f_j-f_i|}\right), \quad \text{with } s = \frac{d^*}{s_1 \min(f_i, f_j) + s_2}. \end{aligned} \tag{5.6}$$

Extension for dissonance between arbitrary tones This approach can be further modified to get the dissonance values for intervals of arbitrary tones or between different timbres. Two complex tones F and G are assumed, consisting of N, respectively M partial tones with frequencies $f_{F,0} < f_{F,1} < \cdots < f_{F,N-1}$ and $f_{G,0} < f_{G,1} < \cdots < f_{G,M-1}$ with their amplitudes $v_{F,0}, v_{F,1}, \ldots, v_{F,N-1}$ and $v_{G,0}, v_{G,1}, \ldots, v_{G,M-1}$. The dissonance is now defined as

$$D = D_F + D_G + \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} v_{ij}\left(e^{-as|f_{G,j}-f_{F,i}|} - e^{-bs|f_{G,j}-f_{F,i}|}\right), \tag{5.7}$$

hence the dissonance is the sum of the dissonance of the timbres itself as well as between the partials of the timbres.

To illustrate this, the dissonance curve of two complex tones with 7 harmonic partials and unit amplitude is calculated and presented in fig. 5.2 to show consonant intervals.

Simplification for timbres and DFT spectrum analysis The calculation for the dissonance of a timbre itself as well as a whole N-point DFT transformed spectrum is of interest in this thesis. An approach was chosen, where each of the magnitude spectrogram bins was interpreted as a partial of a timbre like presented in eq. (5.6). Therefore, the dissonance between each bin is calculated and summed. Due to the structure of the data, some simplifications can be done to reduce computational complexity. As

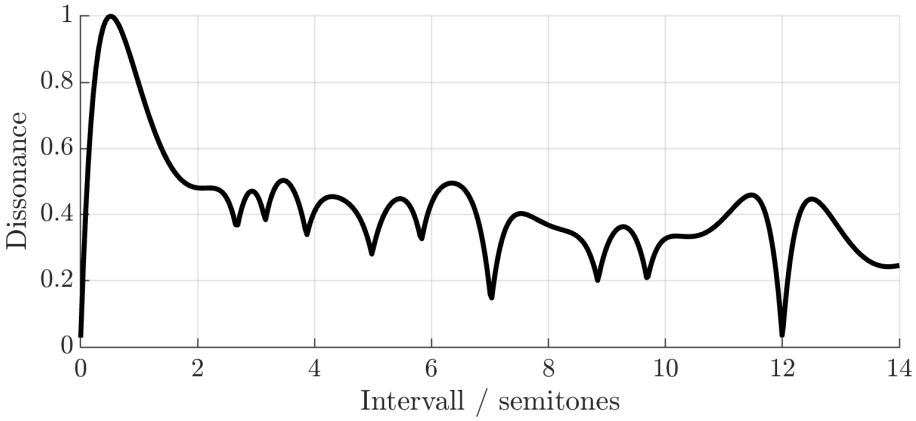


Figure 5.2 – Dissonance as function of an interval in semitones for a base frequency of $500\,\mathrm{Hz}$ for timbres with 7 partial tones with union amplitude.

each bins are compared to another, most of the values are computed twice, for example $d(f_0, f_1, v_0, v_1) = d(f_1, f_0, v_1, v_0)$. Also the dissonance between one bin compared to itself must be zero because

$$d(f_i, f_i, v_i, v_i) = v_i^2 \left(e^{-as(f_i - f_i)} - e^{-bs(f_i - f_i)}\right) = v_i^2 \left(e^0 - e^0\right) = 0\,. \tag{5.8}$$

When using both properties, computational effort can be reduced by more than half, namely to

$$D_F = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} v_{ij} \left(e^{-as(f_j - f_i)} - e^{-bs(f_j - f_i)}\right)\,, \quad \text{with } s = \frac{d^*}{s_1 f_i + s_2}\,. \tag{5.9}$$

Computational complexity could be further reduced, when using the property that dissonance between two partials with a frequency difference of more than the critical bandwidth approaches 0. Hence the dissonance between two tones far enough apart [3] must not be calculated.

Gain normalization As the calculation of the dissonance includes a factor of the product of the amplitudes in eq. (5.4), the results are obviously dependent on the level of the signal. For example, if the level of timbres to be analyzed is increased by factor two, the resulting dissonance is increased by factor four. This property does have no impact on the use of Sethares' method for the analytical assessment of timbres, but on a real-time analysis based on a DFT. Therefore, a normalization of the amplitude coefficient by means of

3. As approximation, for frequencies above $250\,\mathrm{Hz}$ a frequency difference of an octave is sufficient [Set93].

a squared ℓ_2-norm is proposed with

$$\tilde{v}_{ij} = \frac{v_i v_j}{\epsilon + \sum_{k=0}^{N-1} |v_k|^2}\,, \tag{5.10}$$

which compensates the influence of the input signal level. An additional regularization parameter ϵ can be used to avoid large values for small magnitudes.

The proposed analysis, based on a CQT transform, is displayed in fig. 5.3. One can see, that the sudden jumps of dissonance correspond well with the start of bells, especially at later moments. The increased dissonance at $9\,\mathrm{s}$ and $20\,\mathrm{s}$ to $25\,\mathrm{s}$ can be explained by noticeable wind noise at this moments in the recording.

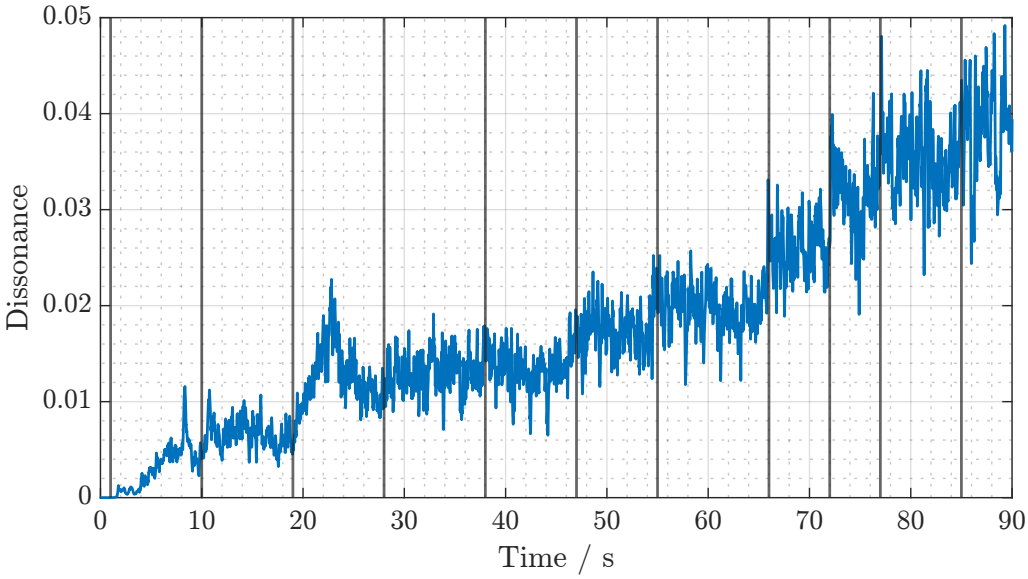


Figure 5.3 – Dissonance of the bells of Graz Cathedral and Mausoläum. Vertical lines indicate the start of another bell.

Chapter 6

Implementation

The algorithms proposed throughout chapter 3, section 4.3.3 and chapter 5 were also implemented in form of a real-time capable application. At first, recordings of the exact location are analyzed offline to find optimal parameter settings. The hereby generated templates and parameter-sets can be exported and imported into a real-time capable application, based on an already existing CQT Analyzer [Hol+20a].

6.1 Offline calculations

In order to find a capable set of parameters, a reference recording is analyzed offline using using the software MATLAB in combination with the NMF-toolbox [Lóp+19] and a reference implementation of the CQT [Hol+20b]. Various settings can be changed, e.g. the CQT parameters, the used cost function or the NMFD dictionary length. The number of components is typically known a priori. Once a suitable variable- and template-set is found, they can be exported out of MATLAB in form of a JSON-file [17].

6.2 Real-time implementation

The online estimation of consonance and activations is implemented in the form of a VST plugin or standalone-application based on the JUCE framework [Sto20]. The JSON-file containing templates and parameters, calculated in section 6.1, can be imported. As its structure is stated in detail in [Hol+20b], only a brief summary is provided in this report.

6.2.1 CQT Analyzer

The CQT Analyzer application consists of several classes:

PluginProcessor The PluginProcessor can be seen as the central point of the program. Inputs and outputs are defined here, this includes both audio signals as well as OSC data. All parameters are stored here, and different necessary objects are created, including the PluginEditor and the CQTThread.

PluginEditor This method is relevant for the graphical interface and provides controls for the user. The PluginProcessor listens for changes at the user controls.

CQTThread In this method, the actual CQT algorithm is implemented. The CQT-Thread receives the input audio data, forwarded by the PluginProcessor, which are asynchronically processed to the CQT coefficients. The results are stored in a first-in-first-out (FIFO) queue for further processing or visualization.

OverlappingSampleCollector The input data of the CQTThread is buffered using this method, ensuring that data blocks with 50% overlap are available.

BufferQueue The results of the CQT analysis are stored in this objects, resembling FIFO queues. This is necessary due to the asynchronous processing. This objects can be used to transfer data between several objects conveniently.

CQTVisualizer In order to show the resulting CQT coefficients, the CQTVisualizer object is created by the PluginEditor. It paints the coefficients on the appropriate positions.

The general structure of the application is visualized in fig. 6.1.

6.2.2 Calculation of activations

The calculation of the activations for each component happens by means of an NMFD approach, described in section 4.3.3. The method to do so itself is integrated into the CQTThread. An additional queue was introduced, which collects and stores the most recent CQT coefficients. As this approach can be computationally quite extensive, it can be exploited that the data is used for visualization only. Hence, only roughly 25 values per second are needed for a smooth visualization. Therefore, the calculation of the activation

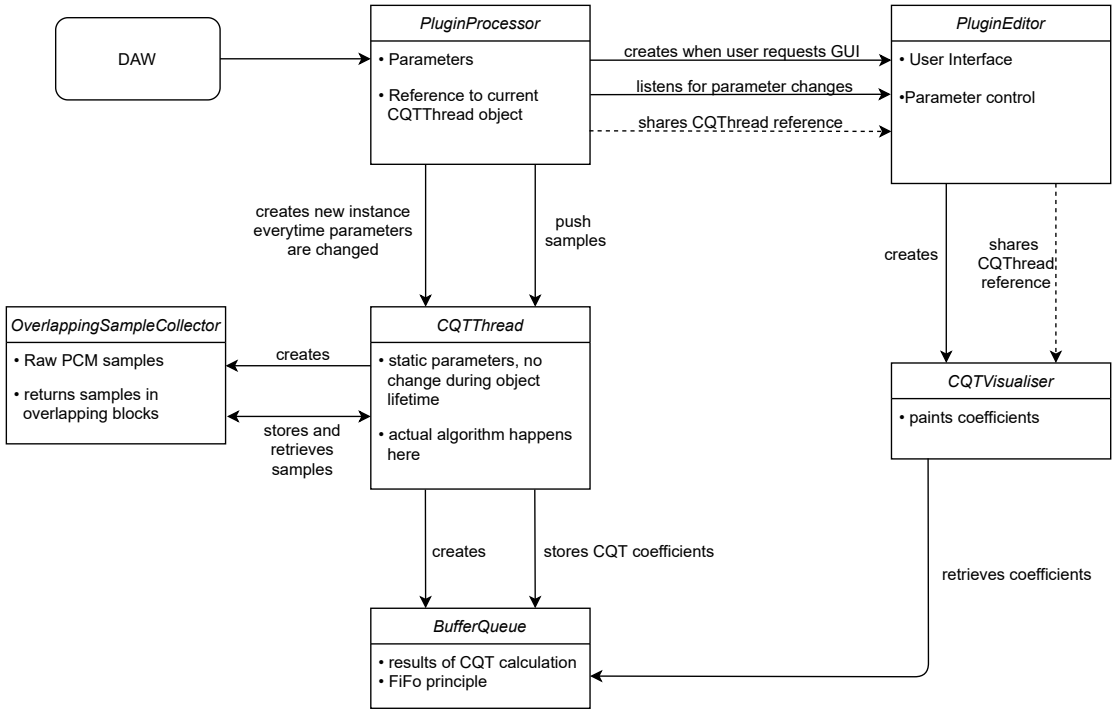


Figure 6.1 – Overview over the program structure. [Hol+20b]

is triggered in an interval so that a frame rate of 25 frames per second can be reached. The calculated values are now stored in a FIFO queue in order to be sent later on as OSC message. An additional normalization, based on a maximum value in the preliminary offline analysis, can be applied to get a result in the range of 0 and 1.

6.2.3 Calculation of local consonance

Also the calculation of the gain-normalized local consonance is part of the CQTThread. Not only the overall consonance is calculated, but also the individual consonance for each bell. As mentioned in section 2.2, the partials of bell sounds have different decay times, therefore the consonance of a single component itself changes over time. In order to do so, the most relevant CQT-bin indices are determined in the offline analysis and passed to the real-time application. The local consonance is now calculated only over those bins. Similar to section 6.2.2, the results are pushed into a FIFO queue in order to send them. Also in this step, an additional normalization using preliminary results can be applied for mapping into the range of 0 and 1.

6.2.4 Output via open sound control (OSC)

To process the data in a visualization application, they have to be transmitted in the first place. A versatile protocol, which allows sending bundled data in real-time as packages via standard network infrastructure is the *open sound protocol* (OSC) [Mat]. The IEM Plug-In template [Rud] already has a built-in OSC routine. The send-interval is set when loading the JSON file, so that a continuous flow of data is guaranteed.

6.2.5 Graphical interface

The graphical user interface of the application is shown in fig. 6.2. On the left hand side, the calculated CQT coefficients are displayed. This visualization can also be turned off for computational savings. On the right hand side, the JSON file can be loaded. A console returns messages if errors occur during loading and displays relevant information if the templates are loaded successfully. The OSC address can be set in the eponymous field.

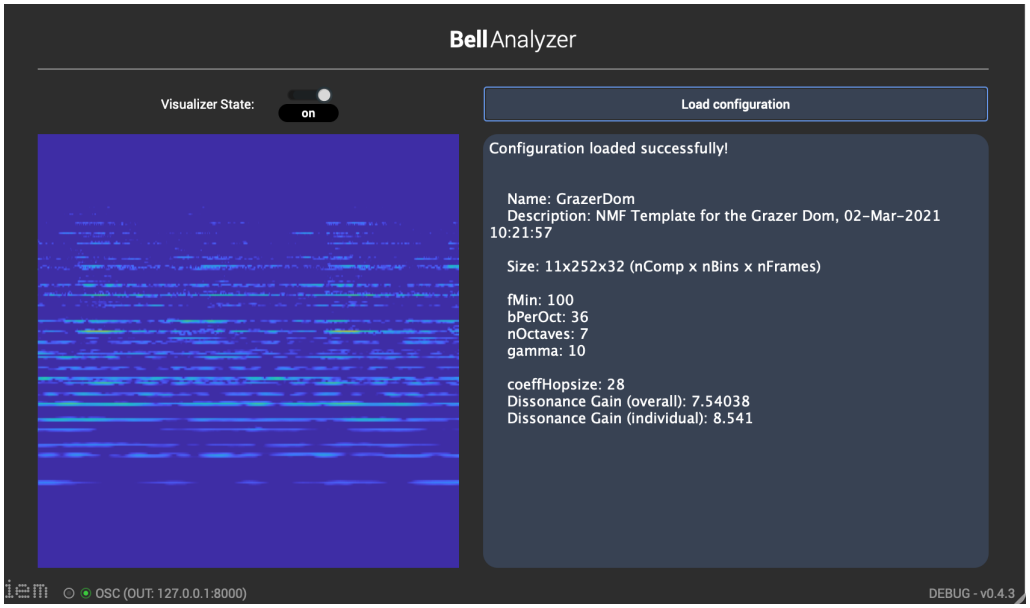


Figure 6.2 – Graphical interface of the application.

6.2.6 Latency

The latency of the application is of course an important factor for its usability. The combined delay of the audio signal analysis and the visualization should not exceed the transit time of sound between bells and spectators for a conclusive visualization. The lower bound for the audio analysis latency can be approximated by the required FFT-size of the CQT analysis, assuming that the NMFD template length is short in comparison to

the returned block of CQT coefficients. Deliberations about the maximal tolerable latency should of course be incorporated into the preliminary offline analysis and parameter choice.

Chapter 7

Visualization

In the course of this project, also a visualization as proof-of-concept and for internal demonstration was created.

7.1 Graphical concept

One of the main goals was to find a visual concept, which blends with (neo-)Gothic architectural aesthetics as well as qualifies for projection. Also a connection between the calculated parameters, western musical elements and ecclesiastical motives should be made. It still has to be comprehensible, plausible and visually appealing to hearing impaired persons.

An element which comes to mind is the Gothic rose window. Each bell is represented by twelve geometric sub-elements, arranged in a circle. The radius and for that reason also the order of the circles correspond to the strike tone. This means, that the bell with highest pitch is located near the center, whereas elements with lower pitch are arranged outwards. This can be seen as reference between the musical domain (twelve half-tones per octave in western music) and elements of Abrahamic religions (e.g. twelve tribes of Israel). The bells can therefore be represented as concentric circles.
The brightness of the circles is altered, depending on the activation. Also a short fade out of the elements is implemented using pd's "line"-object with a duration of $1\,\mathrm{s}$. This results in a vibrant, pulsating graphical representation corresponding to the bell sounds itself.

In future concepts, the decay time of each partial itself, broken down to the chroma [1], can be set individually for each of the twelve sub-elements.

1. Chroma refers in the context of music information retrieval to the pitch using twelve half-tones in western music.

To explore another dimension, the consonance of the sound can be processed in terms of the color of the elements.

7.2 Implementation

The visualization itself was implemented in pure data [Puc] using the "Graphical Environment for Multimedia" (Gem) external [Dan+19].
The incoming OSC messages are parsed and sent to the corresponding elements and sub-elements. The concentric circles are abstractions, consisting of the twelve sub-elements, accordingly arranged and rotated. Therefore, with additional parameters as the radius and an angular offset, the concentric circle objects can be reused with several instances. The sub-elements itself are based on Gem's *square* element.

Excerpts at different times of this conceptual visualization are shown in figs. 7.1 and 7.2.

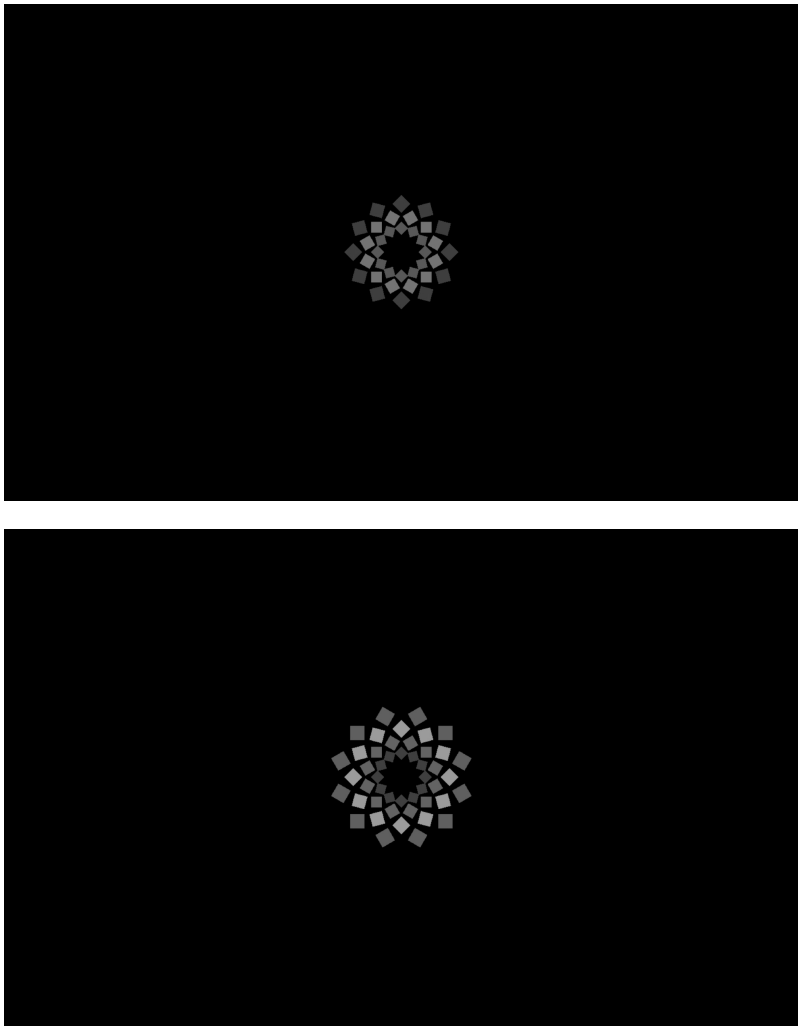


Figure 7.1 – Screenshots of the conceptual visualization.

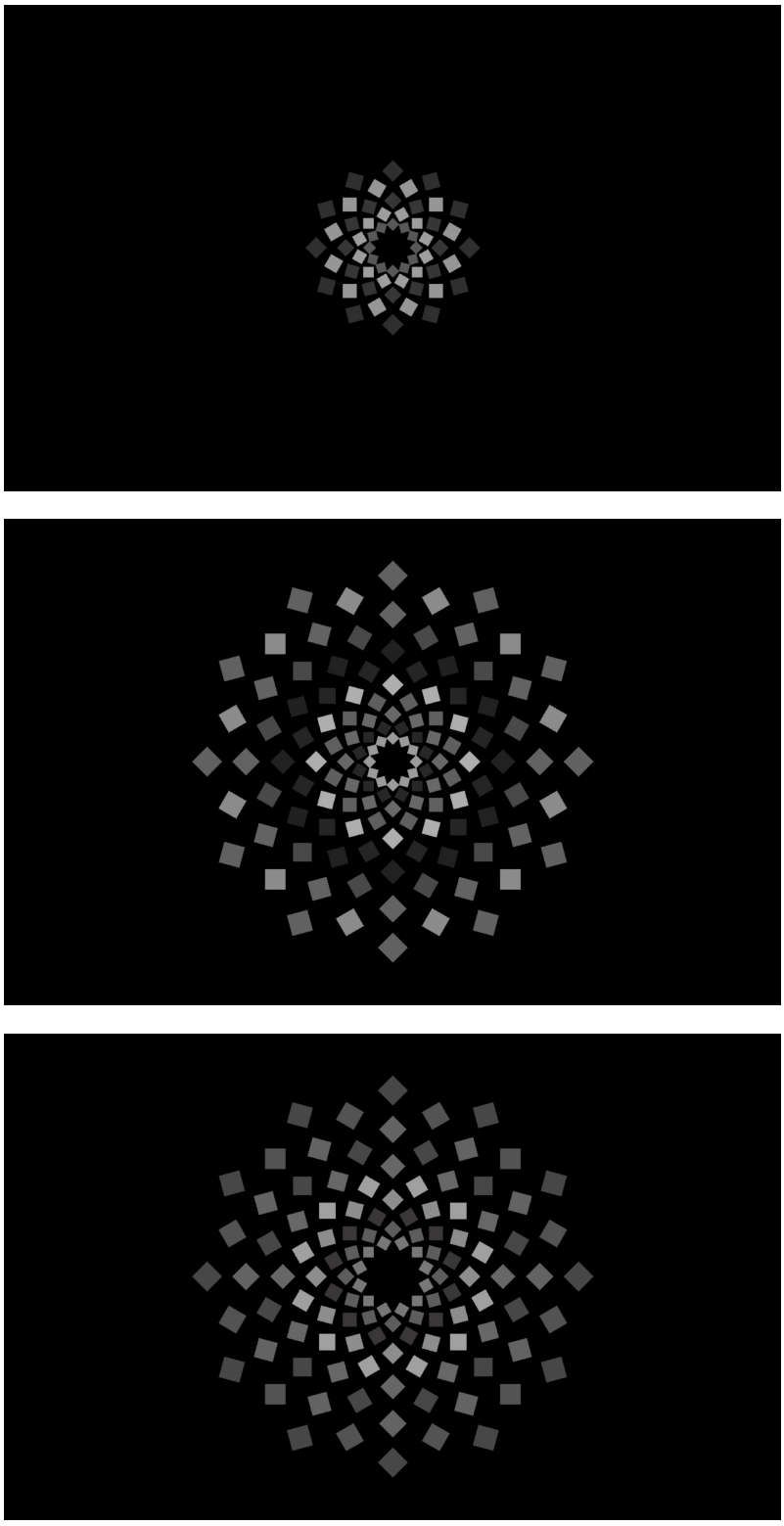


Figure 7.2 – Screenshots of the conceptual visualization.

Chapter 8

Conclusion and outlook

The goal of this project was to find different approaches for analysis of bell sounds. Particular attention was paid to the calculation of activations, harmonic analysis and perceived consonance. The parameters, calculated in real-time, are used as input for a visualization, aimed at hearing impaired persons for the purpose of an inclusive cultural project.
In the course of this work, the acoustic characteristics of bell sounds were presented. The partials are typically not entirely harmonic sounds. Another uncommon attribute are the vastly different decay times of different partial tones.
As for a thorough analysis in time-frequency domain both high frequency resolution as well as high temporal resolution is desired, a discrete Fourier transform was ruled out. An alternative approach is the constant-Q transform, which can meet both demands. With an algorithm based on a fast Fourier transform and a subband technique, the CQT can be calculated efficiently.
A central aspect of this analysis framework is the computation of activations. These can be seen as the excitation pattern, triggering the bells. For different data situation, varying calculation approaches are presented. If recordings for each separate bell are available, the Kullback-Leibler divergence proved as effectve method. This technique does not work for multiple components present on one audio channel. For a limited number of elements, a simple template based approach can work. Further improvements can be observed using templates including their temporal progress, obtained using a non-negative matrix factor deconvolution. This improvement comes at the cost of computational expanses and additional delay. Another parameter discussed is the local consonance. Based on the critical bandwidth theory of human hearing, a parametric model is derived. With simple modifications, it is capable for efficient real-time calculation.
A NMFD based calculation of activation as well as the local consonance model was implemented as real-time capable application as extension to an already existing CQT Analyzer. The calculated data serves as input for a visualization, picking up musical, architectural and religious concepts.

Further improvements could include a more robust estimation of activations in template-based and NMFD methods. In particular stochastic disturbances such as wind noise can influence the results. Also further modification of Sethares' consonance model to an absolute value, independent from the chosen parameters and signal, would be desirable.

Bibliography

[Aka21] Akademie Graz. *Kultur Inklusiv*. 2021. URL: https://www.akademie-graz.at/cms/cms.php?pageName=2&terminId=546 (visited on 12/25/2021).

[ANP11] Fabrizio Argenti, Paolo Nesi, and Gianni Pantaleo. "Automatic Transcription of Polyphonic Music Based on the Constant-Q Bispectral Analysis". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.6 (2011), pp. 1610–1630. ISSN: 1558-7916. DOI: 10.1109/tasl.2010.2093894.

[BS03] Juan Pablo Bello and Mark Sandler. "Phase-Based Note Onset Detection for Music Signals". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 5. Hong Kong, China, 2003, pp. 441–444. ISBN: 0-7803-7663-3. DOI: 10.1109/icassp.2003.1200001.

[Bro06] Paul M. Brossier. "Automatic Annotation of Musical Audio for Interactive Applications". 2006.

[Bro91] Judith C. Brown. "Calculation of a Constant Q Spectral Transform". In: *The Journal of the Acoustical Society of America* 89.1 (1991), pp. 425–434. ISSN: 0001-4966. DOI: 10.1121/1.400476.

[BP92] Judith C. Brown and Miller S. Puckette. "An Efficient Algorithm for the Calculation of a Constant Q Transform". In: *The Journal of the Acoustical Society of America* 92.5 (Feb. 1992), pp. 2698–2701. ISSN: 0001-4966. DOI: 10.1121/1.404385.

[Chl87] Ernst Florens Friedrich Chladni. *Entdeckungen Über Die Theorie Des Klanges*. Leipzig: Weidmanns Erben und Reich, 1787. URL: http://www.deutschestextarchiv.de/book/show/chladni_klang_1787.

[Cic+09] Andrzej Cichocki et al. *Nonnegative Matrix and Tensor Factorizations*. 1st ed. Chichester, UK: John Wiley & Sons, Sept. 2009. ISBN: 978-0-470-74727-8. DOI: 10.1002/9780470747278.

[Dan+19] Mark Danks et al. *Gem — Pd Community Site*. Version 0.94. Mar. 15, 2019. URL: http://gem.iem.at/ (visited on 12/25/2021).

[DCL10] Arnaud Dessein, Arshia Cont, and Guillaume Lemaitre. “Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence”. In: *ISMIR - 11th International Society for MusicInformation Retrieval Conference,* 2010, pp. 489–494.

[Dri+13] Jonathan Driedger et al. “Score-Informed Audio Decomposition and Applications”. In: *Proceedings of the 21st ACM International Conference on Multimedia*. 2013, pp. 541–544. ISBN: 978-1-4503-2404-5. DOI: 10.1145/2502081.2502143.

[Dux+03] Chris Duxbury et al. “Complex Domain Onset Detection for Musical Signals”. In: *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03)*. London, UK, 2003.

[EK01] Shinto Eguchi and Yutaka Kano. “Robustifing Maximum Likelihood Estimation by Psi-Divergence”. In: (Jan. 2001).

[FBD09] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. “Nonnegative Matrix Factorization with the Itakura-Saito Divergence: With Application to Music Analysis”. In: *Neural Computation* 21.3 (2009), pp. 793–830. ISSN: 0899-7667. DOI: 10.1162/neco.2008.04-08-771. pmid: 18785855.

[Fle97] Helmut Fleischer. *Glockenschwingungen*. Ed. by Hugo Fastl Helmut Fleischer. Vol. 1/97. Beiträge Zur Vibro- Und Psychoakustik. Neubiberg, Deutschland: UniBw, 1997.

[FFS07] Helmut Fleischer, Hugo Fastl, and Martin Sattler. “Wann Klingt Ein Glockenklang Nach Kirchenglocke?” In: *DAGA 2007*. Stuttgart, 2007, pp. 241–242.

[GM90] Brian R. Glasberg and Brian C. J. Moore. “Derivation of Auditory Filter Shapes from Notched-Noise Data”. In: *Hearing Research* 47.1-2 (1990), pp. 103–138. ISSN: 0378-5955. DOI: 10.1016/0378-5955(90)90170-t. pmid: 2228789.

[HM03] Stephen Hainsworth and Malcolm Macleod. “Onset Detection in Musical Audio Signals”. In: *Proceedings of the International Computer Music Conference (ICMC)*. Singapore, 2003, pp. 163–166. URL: https://www.researchgate.net/profile/Malcolm_Macleod/publication/2921601_Onset_Detection_in_Musical_Audio_Signals/links/00b4952469d55b8d51000000.pdf.

[Hel63] Hermann von Helmholtz. *Die Lehre von Den Tonempfindungen Als Physiologische Grundlage Für Die Theorie Der Musik*. Braunschweig: Friedrich Vieweg und Sohn, 1863.

[Hol+13] Nicki Holighaus et al. “A Framework for Invertible, Real-Time Constant-Q Transforms”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.4 (2013), pp. 775–785. ISSN: 1558-7916. DOI: 10.1109/tasl.2012.2234114.

[Hol+20a] Felix Holzmüller et al. “Computational Efficient Real-Time Capable Constant-Q Spectrum Analyzer”. In: *AES 148th Convention*. Online, May 2020. URL: http://www.aes.org/e-lib/browse.cfm?elib=20805.

[Hol+20b] Felix Holzmüller et al. *Computational Efficient Real-Time Capable Constant-Q Spectrum Analyzer*. Documentation. Mar. 4, 2020, p. 14. URL: https://git.iem.at/audioplugins/cqt-analyzer (visited on 12/23/2021).

[17] *Information Technology - The JSON Data Interchange Syntax*. Standard ISO/IEC 21778:2017(E). ISO/IEC, Nov. 2017, Vernier, Geneva, Switzerland. URL: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/16/71616.html (visited on 01/06/2022).

[Kau02] Ismo Kauppinen. “Methods for Detecting Impulsive Noise in Speech and Audio Signals”. In: *International Conference on Digital Signal Processing*. Vol. 14. 2002, pp. 967–970. ISBN: 0-7803-7503-3. DOI: 10.1109/icdsp.2002.1028251.

[Kli04] Ingmar-Leander Klich. “Automatische Erkennung von Onsets in Musiksignalen Zur Steuerung von Beattracking-Systemen”. Aug. 2004.

[Kom07] Raul Kompass. “A Generalized Divergence Measure for Nonnegative Matrix Factorization”. In: *Neural computation* 19.3 (2007), pp. 780–791. DOI: 10.1162/neco.2007.19.3.780.

[Kru03] Felix Krueger. “Differenztöne Und Konsonanz”. In: *Archiv f. d. gesamte Psychologie* 1 (1903), pp. 205–275.

[Kur+21] Astrid Kury et al. *Grazer Leitfaden Für Inklusive Kultur*. Graz: Akademie Graz, InTaKT Festival, 2021. ISBN: 978-3-200-07608-2. URL: https://www.kulturjahr2020.at/wp-content/uploads/2021/05/AKADEMIE_GRAZ_Leitfaden_inklusive_Kultur.pdf (visited on 12/25/2021).

[LE20] Imre Lahdelma and Tuomas Eerola. “Cultural Familiarity and Musical Expertise Impact the Pleasantness of Consonance/Dissonance but Not Its Perceived Tension”. In: *Sci Rep* 10.1 (Dec. 2020), p. 8693. ISSN: 2045-2322. DOI: 10.1038/s41598-020-65615-8.

[LS99] Daniel D. Lee and H. Sebastian Seung. “Learning the Parts of Objects by Non-Negative Matrix Factorization”. In: *Nature* 401.6755 (1999), pp. 788–791. ISSN: 0028-0836. DOI: 10.1038/44565. pmid: 10548103.

[LS01] Daniel D. Lee and Sebastian H. Seung. "Algorithms for Non-Negative Matrix Factorization". In: *Advances in Neural Information Processing Systems* 13 (2001). Ed. by T. Leen, T. Dietterich, and V. Tresp, pp. 556–562. URL: https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

[Lóp+19] Patricio López-Serrano et al. "NMF Toolbox: Music Processing Applications of Nonnegative Matrix Factorization". In: 22 (2019). URL: https://www.audiolabs-erlangen.de/content/resources/MIR/00-2019_TutorialFMP_ISMIR/2019_LopezSerranoDOM_NMF_DAFx.pdf.

[Mat] Adrian Freed Matt Wright. *Open Sound Control*. URL: https://opensoundcontrol.org/.

[PL65] R. Plomp and W. J. M. Levelt. "Tonal Consonance and Critical Bandwidth". In: *The Journal of the Acoustical Society of America* 38.4 (1965), pp. 548–560. ISSN: 0001-4966. DOI: 10.1121/1.1909741. pmid: 5831012.

[Puc] Miller S. Puckette. *Pure Data — Pd Community Site*. Version 0.51.3. URL: https://puredata.info/ (visited on 12/25/2021).

[Ray90] John William Strutt Rayleigh 3rd Baron. "On Bells". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 29.176 (1890).

[Rud] Daniel Rudrich. *IEM Plug-in Suite*. URL: https://plugins.iem.at/ (visited on 12/23/2021).

[Sch+14] Christian Schörkhuber et al. "A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution". In: vol. 53. AES International Conference. 2014, pp. 1–8.

[Set93] William A. Sethares. "Local Consonance and the Relationship between Timbre and Scale". In: *The Journal of the Acoustical Society of America* 94.3 (1993), pp. 1218–1228. ISSN: 0001-4966. DOI: 10.1121/1.408175.

[Sma04] Paris Smaragdis. "Non-Negative Matrix Factor Deconvolution; Extracation of Multiple Sound Sources from Monophonic Inputs". In: *International Congress on Independent Component Analysis and Blind Signal Separation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 494–499. ISBN: 978-3-540-23056-4. DOI: 10.1007/978-3-540-30110-3_63.

[Sot94] Roland Sottek. "Gehörgerechte Rauhigkeitsberechnung". In: *Fortschritte der Akustik - Plenarvorträge und Fachbeiträge der 20. Deutschen Jahrestagung für Akustik*. Deutsche Jahrestagung für Akustik. Vol. 20. Dresden, 1994, pp. 1197–1200. URL: https://global.head-acoustics.com/downl

oads/publications/hearing_related/Daga1994_Gehoergerechte_Rauigkeit.pdf.

[Sto20] Julian Storer. *Juce-Framework/JUCE*. Version 5.4.7. JUCE, Feb. 10, 2020. URL: https://github.com/juce-framework/JUCE (visited on 01/06/2022).

[Vel+11] Gino Angelo Velasco et al. "Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames". In: *Proc. of the 14th International Conference on Digital Audio Effects (DAFx-11))*. Vol. 14. Sept. 2011, pp. 93–99.

[Wer04] Jörg Wernisch. "Untersuchungen an Kirchenglocken". 2004.

[Wik20] Wikimedia Commons. *File:Parts of a Bell.Svg — Wikimedia Commons, the Free Media Repository*. 2020. URL: https://commons.wikimedia.org/w/index.php?title=File:Parts_of_a_Bell.svg&oldid=439461045.

[WR17] Sean U. N. Wood and Jean Rouat. "Real-Time Speech Enhancement with GCC-NMF". In: *Proc. Interspeech 2017*. 2017, pp. 2665–2669. DOI: 10.21437/Interspeech.2017-1458.

[YB78] James Youngberg and Steven Boll. "Constant-Q Signal Analysis and Synthesis". In: *ICASSP'78. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 3. IEEE. 1978, pp. 375–378.

List of Figures

List of Tables