Universidad de la República
Facultad de Ingeniería

Audio Source Separation Techniques Including Novel Time-Frequency Representation Tools

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Pablo Cancela

en cumplimiento parcial de los requerimientos para la obtención del título de Doctor en Ingeniería Eléctrica.

Director de Tesis

Guillermo Sapiro ..... Duke University

Tribunal

Sergio Lima ..... Universidade Federal do Rio de Janeiro
Juan Pablo Bello ..... New York University
Luis Weruaga ..... Kahlifa University
Federico Lecumberry ..... Universidad de la República
Pablo Musé ..... Universidad de la República

Director Académico

Gregory Randall ..... Universidad de la República

Montevideo
Diciembre 2015

UNIVERSIDAD DE LA REPÚBLICA

Abstract

Facultad de Ingeniería
Instituto de Ingeniería Eléctrica

Doctor of Philosophy

Audio Source Separation Techniques Including Novel Time-Frequency Representation Tools.

by Pablo Cancela

The thesis explores the development of tools for audio representation with applications in Audio Source Separation and in the Music Information Retrieval (MIR) field.

A novel constant Q transform was introduced, called IIR-CQT. The transform allows a flexible design and achieves low computational cost.

Also, an independent development of the Fan Chirp Transform (FChT) with the focus on the representation of simultaneous sources is studied, which has several applications in the analysis of polyphonic music signals.

Different applications are explored in the MIR field, some of them directly related with the low-level representation tools that were analyzed. One of these applications is the development of a visualization tool based in the FChT that proved to be useful for musicological analysis . The tool has been made available as an open source, freely available software.

The proposed Transform has also been used to detect and track fundamental frequencies of harmonic sources in polyphonic music. Also, the information of the slope of the pitch was used to define a similarity measure between two harmonic components that are close in time. This measure helps to use clustering algorithms to track multiple sources in polyphonic music.

Additionally, the FChT was used in the context of the Query by Humming application. One of the main limitations of such application is the construction of a

search database. In this work, we propose an algorithm to automatically populate the database of an existing Query by Humming, with promising results.

Finally, two audio source separation techniques are studied. The first one is the separation of harmonic signals based on the FChT. The second one is an application for which the fundamental frequency of the sources is assumed to be known (Score Informed Source Separation problem).

Contents

Abbreviations

CQT	**C**onstant **Q**uality-Factor **T**ransform
IR-CQT	**I**nfinite **I**mpulse **R**esponse **C**onstant **Q**uality **T**ransform
MRFFT	**M**ulti **R**esolution **F**ast **F**ourier **T**ransform
STFT	**S**hort **T**ime **F**ourier **T**ransform
FChT	**F**an **Ch**irp **T**ransform
STFChT	**S**hort **T**ime **F**an **Ch**irp **T**ransform
GMM	**G**aussian **M**ixture **M**odels
QBH	**Q**uery **B**y **H**umming
DFT	**D**iscrete **F**ourier **T**ransform
BQT	**B**ounded **Q**uality **T**ransform
LTV	**L**inear **T**ime **V**ariant
SSR	**S**teady **S**tate **R**esponse
WVD	**W**igner **V**ille **D**istribution
SCSS	**S**ingle **C**hannel **S**ource **S**eparation
NMF	**N**onnegative **M**atrix **F**actorization
MAP-EM	**M**aximum **A**-**P**osterori **E**xpectation **M**aximization
PCA	**P**rincipal **C**omponent **A**nalysis
MIREX	**M**usic **I**nformation **R**etrieval **EX**change
LDTW	**L**ocal **D**ynamic **T**ime **W**arping
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
SVM	**S**upport **V**ector **M**achines
Gaussian **RBF**	Gaussian **R**adial **B**asis **F**unction
LFP	**L**ow **F**requency **P**ower
MRR	**M**ean **R**eciprocal **R**ank

SDR	**S**ource to **D**istortion **R**atio
SAR	**S**ource to **A**rtifacts **R**atio
SIR	**S**ource to **I**nterference **R**atio

Chapter 1

Introduction

1.1 Introduction

Sound plays an important role in human communication, including the exchange of information, emotions and being a basic and fundamental way of socialization. One of the main ways of communication is speech, in which the message is modulated to fit in a relatively narrow bandwidth in an efficient an robust way. Also, a great variety of animals communicate through sound in many different frequency bands, usually adapted to their environment. Sound communication for them also involves many different aspects of their life to situate each other, express their mood, protect an area, socialize, among many others.

Sound is at the heart of the most essential ways of the production of art, as music has formed part of almost every culture. Nowadays, with the popularization of high performance devices such as smart phones, the access, processing and analysis of multimedia content pose challenging problems. For instance, applications such as recognizing a musical piece by a fragment or by a short singing or humming query, are becoming accessible for everyday use. Also, a graphical representation of the contents of a music excerpt assist academic studies that use quantitative information to support arguments of musical aspects of an interpretation. Many automatic music analysis algorithms such as those designed to extract the melody of a musical piece or track multiple instrument pitches in a polyphonic mix are the input for other content related applications. In order to address these problems it is important to have an accuratee representation of audio both in time and frequency simultaneously, what poses different challenges.

There are limits in the maximum resolution that can be simultaneously obtained in time and frequency. These bounds are outlined by the uncertainty principle in Signal Analysis, which states that as time and frequency are dual spaces related by the Fourier Transform, a signal in time $s(t)$ cannot simultaneously have an arbitrary small duration T and bandwidth B, which represent the dispersion of energy defined by the equations

$$T^2 = \int (t - < t >)^2 |s(t)|^2 dt$$

$$B^2 = \int (w - < w >)^2 |S(w)|^2 dw$$

where $s(t)$ and $S(w)$ are the expressions of the signal s in the time and frequency domains, and the time and frequency centers of the signals are defined by,

$$< t >= \int t \cdot |s(t)| dt$$

$$< w >= \int w \cdot |S(w)| dw.$$

Then, the uncertainty principle states that

$$T \cdot B \geq \frac{1}{2}. \tag{1.1}$$

This can be easily proved based on the definition of the Fourier Transform and the Cauchy–Shwartz inequality (see [1]).

The classical general purpose time–frequency representation is the so-called Short Time Fourier Transform (STFT), in which overlapped windows are applied in the time domain to generate a two dimensional map that represents the components for different time–frequency locations. Another uncertainty principle can be formulated for the representation of signals with the STFT. In this case, the signal is windowed at each frame time, producing a new signal with limited duration. Given the uncertainty principle of Equation 1.1, there is a lower bound for the bandwidth resolution that depends not only on the signal itself but also in the window's properties. If the duration of an event is shorter than the width of the window, the resulting bandwidth is not going to be significantly changed, and a good resolution in frequency is achieved. Conversely, if the event is longer than the window's duration, the representation will blur the information in the frequency domain. The same result can be obtained for the time resolution based on the bandwidth of the window and the bandwidth of signal to be represented. In this

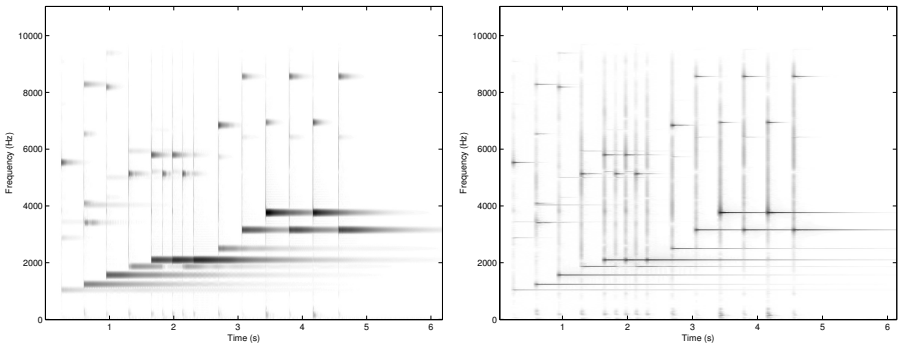


FIGURE 1.1: Example of the representation of sounds with different window lengths. The STFT of a piece of audio with glockenspiel sounds is represented, in which an impulsive part and a tonal part can be observed for each note. In the image of the left a short time Hanning window of 12 ms was used, and in the right the Hanning window is 93 ms long.

case, if the bandwidth of the signal is bigger than the window bandwidth, the representation in time is blurred. There is a trade-off between the resolution that can be achieved in each dimension that is basically defined by the size and the shape of the analysis window. Typical choices of windows that concentrate as much energy as possible in both domains are Hamming, Hanning and Blackman windows which have a limited support in time and optimize different properties of the side lobes in the frequency domain.

When a priori knowledge of the characteristics of the signal to be represented is available, the trade-off between time and frequency resolution can be adjusted to optimize its representation. For instance, if drum sounds are to be studied, a short time window will properly represent the events in time and give as much information as it can be extracted from the spectrum content of the events. On the other hand, if notes with constant pitch are the main content of a musical piece, a longer window will probably represent more appropriately the spectral content while not putting too much stress in representing the exact instant of the onset of the notes with unnecessary accuracy, as it may be actually diffuse.

Figure 1.1 shows an example of glockenspiel sounds with components that are very local in time at their onset, and components that have a stationary behavior. The STFT is calculated for two different window lengths. A short Hanning window, for which the impulsive events that are very local in time appear sharply represented, and the stationary components are diffuse. Conversely, for a long Hanning analysis window, the onset components are blurry, and the frequency of the stationary components exhibit a much better resolution.

The frequency components of a Discrete Fourier Transform (DFT) are equally spaced and have a constant resolution. However, in polyphonic music a higher frequency resolution is needed in the low and mid frequencies where there is a higher density of harmonics. On the other hand, frequency modulation gets stronger as the number of harmonic is increased, requiring shorter windows for improved time resolution. Thus, a multi-resolution spectral representation is highly desired for the analysis of music signals. In addition, computational cost is a critical issue in real time or demanding applications so efficient algorithms are often needed.

In this context several techniques have been proposed to circumvent the conventional linear frequency and constant resolution of the DFT. The constant-Q transform (CQT) [2] is based on a direct evaluation of the DFT but the channel bandwidth Δf_k varies proportionally to its center frequency f_k, in order to keep constant its quality factor $Q = f_k/\Delta f_k$ (as in Wavelets). Central frequencies are distributed geometrically, to follow the equal tempered scale used in Western music, in such a way that there are two frequency components for each musical note (although higher values of Q provide a resolution beyond the semitone). Direct evaluation of the CQT is very time consuming, but fortunately an approximation can be computed efficiently taking advantage of the Fast Fourier Transform (FFT) [3].

Various approximations to a constant-Q spectral representation have also been proposed. The bounded-Q transform (BQT) [4] combines the FFT with a multirate filterbank. Octaves are distributed geometrically, but within each octave, channels are equally spaced, hence the log representation is approximated but with a different number of channels per octave. Note that the quartertone frequency distribution, in spite of being in accordance with Western tuning, can be too scattered if instruments are not perfectly tuned, exhibit inharmonicity or are able to vary their pitch continuously (e.g. glissando or vibrato). A new version of the BQT with improved channel selectivity was proposed in [5] by applying the FFT structure but with longer kernel filters, a technique called Fast Filter Bank. An approach similar to the BQT is followed in [6] as a front-end to detect melody and bass line in real recordings. Also in the context of extracting the melody of polyphonic audio, different time–frequency resolutions are obtained in [7] by calculating the FFT with different window lengths. This is implemented by a very efficient algorithm, named the Multi-Resolution FFT (MR FFT), that combines elementary transforms into a hierarchical scheme.

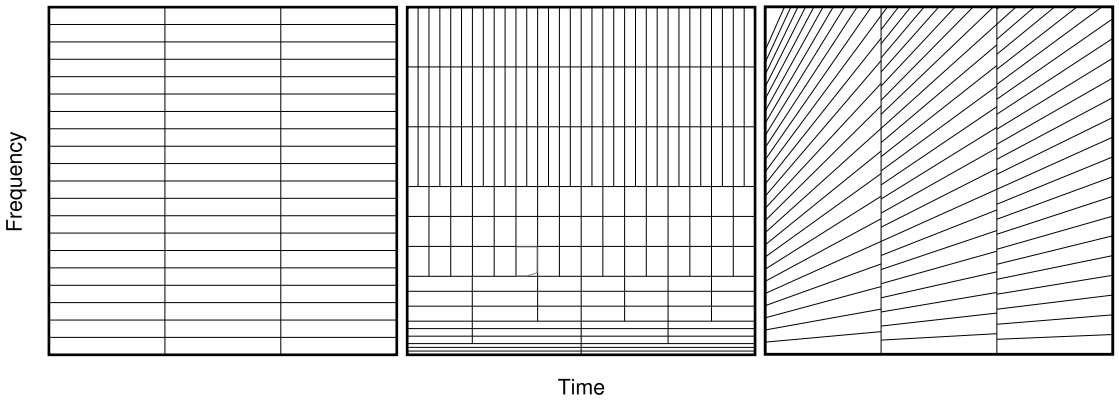


FIGURE 1.2: Time–frequency tiling and resolution compromise for different Representation Transforms. Fom left to right: STFT, CQT and a Chirp based representation.

We focus on multi-resolution spectral analysis algorithms for music signals based on the FFT. Two previously devised efficient algorithms that exhibit different characteristics are reviewed, namely, the efficient CQT [3] and the MR FFT [7]. The former is more flexible regarding Q design criteria and frequency channel distribution while the latter is more efficient at the expense of design constrains. These algorithms are compared with a new proposal based on the Infinite Impulse Response (IIR) filtering of the FFT (IIR CQT), that in addition to its simplicity shows to be a good compromise between design flexibility and reduced computational effort.

In all the mentioned Transforms, the uncertainty principle considers the energy marginalized in the time and frequency dimensions. For signals such as chirps, in which the frequency changes with time, the compromise between time and frequency resolution is degraded, and the product of duration and bandwidth takes a value that is sensitively larger than the minimum stated by the uncertainty principle. This lack of resolution can be diminished if we marginalize the energy in a direction that is parallel to the chirp. As a counterpart there will be a loss of resolution when the marginalization of the energy in this new direction is used to represent a stationary signal. In this way, additional information on properties of the signals can be exploited to improve their time–frequency representation, but it should be used appropriately. Figure 1.2 shows a schematic of the tiling for different transforms in the time–frequency plane that show the resolution trade-off at different frequency ranges.

Most real signals (for instance, music signals) are non-stationary by nature. Moreover, usually an important part of the information of interest has to do with the non stationarity which are associated with particular events such as beginning and

end of notes, modulations, drifts among others. For this reason, time–frequency representations for the analysis of signals whose spectral content varies in time is an active field of research in signal processing [8]. The representation is commonly adapted to the signal in order to enhance significant events so as to facilitate the detection, estimation or classification. An alternative goal is to obtain a sparse representation for compression or denoising. In some cases the elements of the sparse representation become associated with salient features of the signal thus also providing feature extraction [9].

Precisely representing frequency modulated signals, like singing voice, is a challenging problem in signal processing. Many time–frequency transforms can be applied for this purpose. The most popular quadratic time–frequency representation is the Wigner–Ville Distribution (WVD), which offers good time–frequency localization but suffers from interfering cross-terms. Several alternatives were proposed to attenuate the interferences such as the Smoothed Pseudo WVD and other Cohen class distributions [10], but with the side effect of resolution loss due to the smoothing. A different approach to perform the analysis is considering the projection over frequency modulated sinusoids (chirps), in order to obtain a non-Cartesian tiling of the time–frequency plane that closely matches the pitch change rate. Among the chirp-based transforms, the Chirplet Transform [11] and the Fractional Fourier Transform [12] involve the scalar product between the signal and linear chirps (linear FM), and can reach optimal resolution for a single component linear chirp. However, many sounds present in music (e.g. voice) have an harmonic structure, and these transforms are not able to offer optimal resolution simultaneously for all the partials of a harmonic chirp (harmonically related chirps). The Fan Chirp Transform (FChT), originally proposed in [13] was designed to achieve a better representation of nonstationary harmonic speech signals. The FChT can be considered as a time warping followed by a Fourier Transform, which enables an efficient implementation using the FFT.

Although many of these techniques were applied to speech [14], the use of time–frequency representations other than the STFT for music analysis remains rather scarce [9, 15] and in particular the FChT has not been extensively explored for this purpose, but in some few works [16–19]. See figures 3.5 and 3.6 for a comparison of different time–frequency representations applied to a music audio excerpt.

In this work the FChT is applied to the analysis of pitch content in polyphonic music. Besides, it is combined with the CQT to provide time–frequency multiresolution in the fan geometry. The formulation and implementation of the FChT

differ from the one proposed in [13]. The goal of the formulation is to obtain a more acute representation of linear chirps. A positive byproduct is that it also enables the application of arbitrary warpings in order to analyze non-linear chirps straightforwardly. In addition, the implementation is designed with an emphasis on computational cost. Among the various existing approaches for pitch salience computation, the technique adopted in this work is based on gathering harmonically related peaks of the FChT as proposed in [14], which is improved and adapted to deal with polyphonic music.

1.2 Main Contributions

The main contributions of this work are related to the development of tools for audio representation with applications in Audio Source Separation and in the Music Information Retrieval (MIR) field.

A novel constant Q transform was introduced, called IIR-CQT. The transform allows a flexible design and achieves low computational cost.

Also, an independent development of the Fan Chirp Transform (FChT) with the focus on the representation of simultaneous sources was proposed in [20], which has several applications in the analysis of polyphonic music signals. The contributions in this development include:

- the domain where the analysis windows are applied to obtain optimal resolution.
- the suppression of spurious peaks related to harmonics and subharmonics of the represented sources.
- the combination of the multi-resolution representation with the FChT.
- the generalization of the FChT to use a higher order approximation of fundamental frequency contours.
- the use of a sparse representation of harmonic signals using the FChT, with direct applications in source separation.

We think that the exploration, development and improvement of these low-level representation techniques can have a strong impact in the results obtained by

higher level techniques that make use of them. It also brings some insight in understanding the nature of the data algorithms have to manipulate, helping to understand their potential and limitations.

Different applications were explored in the MIR field, some of them directly related with the low level techniques that were analyzed.

One of these applications is the development of a visualization tool based in the FChT that proved to be useful for musicological analysis as reported in [21]. The tool has been made available as an open source, freely available package in Matlab and as a Sonic Visualizer plug-in implemented in C++.

The proposed Transform has also been used to detect and track fundamental frequencies of harmonic sources in polyphonic music. A simple tracking algorithm based on the FChT was submitted to the MIREX Audio Melody Extraction Task [22] and another tracking algorithm based in clustering was reported in [23]. Also, the information of the slope of the pitch was used to define a similarity measure between two harmonic components that are close in time. This measure helps to use clustering algorithms to track multiple sources in polyphonic music.

Tightly linked to another MIR application, the FChT was used in the context of the Query by Humming application. One of the main limitations of such application is the construction of a search database. In this work, an algorithm to automatically populate the database of an existing Query by Humming system was evaluated in [24], with promising results.

Finally, two audio source separation techniques are described. The first one is the separation of harmonic signals based on the FChT. The second one is an application for which the fundamental frequency of the sources is assumed to be known. As reported in [25], sparse codification techniques combined together with Gaussian Mixture Models were applied for the Score Informed Source Separation problem.

1.3 Outline of the Document

The reminder of this document is organized as follows. Chapter 2 presents an overview of different constant quality factor transforms. The IIR-CQT is described in detail and compared with other transforms in terms of design flexibility, computational cost as well as its frequency response.

In Chapter 3 the FChT is described. The F0gram is presented as a way of representing polyphonic content with a harmonic structure. Some examples of how challenging situations such as rapid variations of pitch or simultaneous sources with different fundamental frequency are represented by F0gram are described.

In Chapter 4 some extensions and a deeper analysis of some properties of the FChT are studied. One of these extensions is the combination of the FChT with a CQT, and another one is the use of non linear time warpings. We also present an analysis of some limitations of the technique, that appear as spurious peaks due to the local nature of the processing to calculate the transform.

In Chapter 5, different applications in which the FChT plays an important role as a low level representation are described. The first one is how the FChT can be used as a tool to represent musical content of a piece of audio for its musicological analysis, as described In Section 5.1. Also, a pitch tracking algorithm to extract the main melody from a musical piece is described. This tracking method uses a similarity measure that takes into account information provided by the FChT as is described in Section 5.2.

Two different source separation techniques are described. The first one, based on knowledge of the score of music, use a structured sparse representation and is discussed in Section 6. In the second one, a separation procedure on how to use the FChT for source separation is described in Section 7.2. The FChT naturally represents harmonic sounds in a sparse way, permitting a straightforward way of isolating information from a mix when the correct parameters are chosen. It follows the procedure to re-synthesize the waveform of the isolated source, which can be easily subtracted from the mixture to obtain the rest of the sources in the residual. Finally, in Section 8 a source separation technique based on the FChT was considered to automatically populate a melody database of a Query by Humming System.

Finally, some global conclusions and remarks are presented in Section 9.

Part I

Time–Frequency Representation Tools

Chapter 2

IIR-Constant Q Transform

The Constant Quality-Factor Transform (CQT) is an alternative to the classical STFT which can achieve a larger frequency resolution in the low and mid frequency bands, while representing frequency changes with a good time resolution in the higher bands. As it was mentioned in the Introduction, this is desirable for the representation of polyphonic music, since a higher density of partials is present in the lower bands as well as minor relatively small pitch changes imply sensitive frequency changes of partials in the the high frequency region.

There are different variations of the CQT, with different properties and computational costs. In this Chapter we describe some of the existing transforms, as well as the novel IIR-CQT proposed by the author, that was developed in collaboration with Martín Rocamora and Ernesto López and published in [26]. This chapter is based on that publication, reproducing some of its passages, it includes also some modifications or additions in order to contribute to the structure of this document.

2.1 FIR Q Transform Implementations

2.1.1 Efficient constant Q transform

As stated in [2] a CQT can be calculated straightforwardly based on the evaluation of the DFT for the desired components. Consider the kth spectral component of the DFT:

$$X[k] = \sum_{n=0}^{N-1} w[n]x[n]e^{-j2\pi kn/N}$$

where $w[n]$ is the temporal window function and $x[n]$ is the discrete time signal. In this case the quality factor for a certain frequency f_k equals k, since $Q_k = f_k/\Delta f = f_k N/f_s = k$. This corresponds to the number of periods in the time frame for that frequency. The digital frequency is $2\pi k/N$ and the period in samples is N/k. In the CQT the length of the window function varies inversely with frequency (but the shape remains the same), so that N becomes $N[k]$ and $w[n]$ becomes $w[n,k]$. For a given frequency f_k, $N[k] = f_s/\Delta f_k = f_s Q_k/f_k$. The digital frequency of the kth component is then given by $2\pi Q/N[k]$, the period in samples is $N[k]/Q$ and always Q cycles for each frequency are analyzed. The expression for the kth spectral component of the CQT is then[1],

$$X^{cq}[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} w[n,k]x[n]e^{-j2\pi Qn/N[k]}. \tag{2.1}$$

Direct evaluation of equation (2.1) is time consuming, so an efficient algorithm for its computation has been proposed in [3]. The CQT can be expressed as a matrix multiplication, $X^{cq} = x \cdot T^*$, where x is the signal row vector of length N ($N \geq N[k]\ \forall k$) and T^* is the complex conjugate of the temporal kernel matrix T whose elements $T[n,k]$ are,

$$T[n,k] = \begin{cases} \frac{1}{N[k]} w[n,k]e^{-j2\pi Qn/N[k]} & \text{if } n < N[k] \\ 0 & \text{otherwise.} \end{cases}$$

Computational effort can be reduced if the matrix multiplication is carried out in the spectral domain. Using Parseval's relation for the DFT, the CQT can be expressed as,

$$X^{cq}[k] = \sum_{n=0}^{N-1} x[n]T^*[n,k] = \frac{1}{N}\sum_{k'=0}^{N-1} X[k']K^*[k',k] \tag{2.2}$$

where $X[k']$ and $K[k',\cdot]$ are the DFT of $x[n]$ and $T[n,\cdot]$ respectively. Spectral kernels are computed only once taking full advantage of the FFT. In the case of conjugate symmetric temporal kernels, the spectral kernels are real and near zero over most of the spectrum. For this reason, if only the spectral kernel values larger than a certain threshold are retained, there are few products involved in the evaluation of the CQT (almost negligible compared to the computation of the FFT of $x[n]$).

[1] A normalization factor $1/N[k]$ must be introduced since the number of terms varies with k.

It is important to notice that although the original derivation of the CQT implies a geometrical distribution of frequency bins, it can be formulated using other spacing, for instance a constant separation. In the following, linear spacing is used to put all the compared algorithms under an unified framework.

2.1.2 Multi-resolution FFT

A simple way to obtain multiple time–frequency resolutions is through the explicit calculation of the DFT using different frame lengths. In [7], an efficient technique is proposed where the DFT using several frame lengths is computed by means of the combination of the DFT of small number of samples, called elementary transforms. The idea arises from the observation that a transform of frame length N can be split into partial sums of L terms (assuming $N/L \in \mathbb{N}$),

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}} = \sum_{c=0}^{\frac{N}{L}-1} \sum_{n=cL}^{(c+1)L-1} x[n] e^{-\frac{j2\pi kn}{N}}. \tag{2.3}$$

Each inner sum in Equation 2.3 corresponds to the DFT of length N of a sequence $x_c[n]$, where $x_c[n]$ is an L samples chunk of $x[n]$, time-shifted and zero padded,

$$x_c[n] = \begin{cases} x[n], & cL \leq n < (c+1)L \\ 0, & \text{otherwise.} \end{cases}$$

So, it is possible to obtain a DFT of a frame of size N from N/L elementary transforms of frame size L, defined as

$$X_l[k] = \sum_{n=0}^{L-1} x[n + lL] e^{-j2\pi kn/N},\ l = 0, ..., \frac{N}{L} - 1.$$

To that end, it is enough to add the elementary transforms modified with a linear phase shift to include the time shift of $x_c[n]$, as stated by the shifting theorem of the DFT,

$$X[k] = \sum_{l=0}^{\frac{N}{L}-1} X_l[k] e^{-j2\pi kl/N}. \tag{2.4}$$

This procedure can be generalized to compute the DFT of any frame of length $M = rL$ by adding r elementary transforms ($r = 1, ..., N/L$) in Equation 2.4, which results in N/L possible spectral representations with frequency resolutions $f_s/(rL)$.

The computation of the multi-resolution spectrum from a combination of elementary transforms requires the windowing process to be done by means of convolution product in the frequency domain. Temporal windows of the form

$$w[n] = \sum_{m=0}^{\frac{M}{2}} (-1)^m a_m \cos\left(\frac{2\pi}{M} mn\right) \tag{2.5}$$

are suitable for this purpose because its spectrum has only few non-zero samples. Due to the fact that windowing is applied over zero-padded transforms, it is convenient to consider a periodic time window of the same length of the DFT to avoid the appearance of new non-zero samples of the window spectrum. In this case, the spectrum of a window of the form of Equation 2.5 results in

$$W[k] = \sum_{m=0}^{\frac{M}{2}} (-1)^m \frac{a_m}{2} \left(\delta\left[k - m\frac{N}{M}\right] + \delta\left[k + m\frac{N}{M}\right]\right).$$

For example, in Hann and Hamming windows only a_0 and a_1 are not zero and so its DFT contains solely three non-zero samples. As a counterpart, the restriction that $N/M = N/(rL) \in \mathbb{N}$ must be imposed, reducing the possible number of resolutions to $\log_2(N/L) + 1$.

2.2 IIR Q Transform

2.2.1 FIR/IIR Filterbank

The mentioned methods define a Finite Impulse Response (FIR) filterbank with different impulse responses for different frequencies. The result of applying one of these filters can be regarded as multiplying the frame with a time window, which defines the time/frequency resolution. Variable windowing in time can also be achieved applying an IIR filterbank in the frequency domain. Let us define the k^{th} filter as a first order IIR filter with a pole p_k, and a zero z_k, as,

$$Y_k[n] = X[n] - z_k X[n-1] + p_k Y_k[n-1] \tag{2.6}$$

Its Z transform is given by

$$H_{f_k}(z) = \frac{z - z_k}{z - p_k}.$$

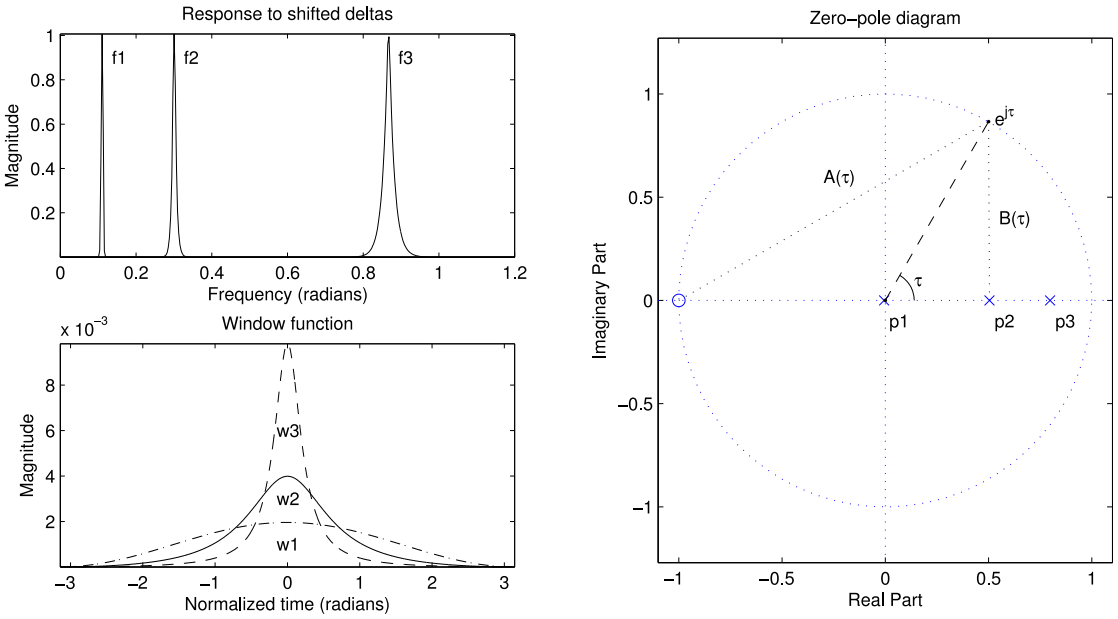


FIGURE 2.1: Zero-Pole diagram and IIR filters responses for three different input sinusoids of frequencies $f_1 = 0.11$, $f_2 = 0.30$ and $f_3 = 0.86$ radians.

Here, $H_{f_k}(z)$ evaluated in the unit circle $z = e^{j\tau}$ represents its time response, with $\tau \in (-\pi, \pi]$ being the normalized time within the frame. A different time window for each frequency bin is obtained by selecting the value of the k^{th} bin as the output of the k^{th} filter.

The design of these filters involves finding the zero and pole for each k such that $w_k(\tau) = |H_{f_k}(e^{j\tau})|$, where $\tau \in (-\pi, \pi]$ and $w_k(\tau)$ is the desired window for the bin k. When a frame is analyzed, it is desirable to avoid discontinuities at its ends. This can be achieved by placing the zero in $\tau = \pi$, that is $z_k = -1$. If we are interested in a symmetric window, $w_k(\tau) = w_k(-\tau)$, the pole must be real. Considering a causal realization of the filter, p_k must be inside the unit circle to assure stability, thus $p_k \in (-1, 1)$. Figure 2.1 shows the frequency and time responses for the poles depicted in the zero-pole diagram.

This IIR filtering in frequency will also distort the phase, so a forward–backward filtering should be used to obtain a zero-phase filter response. Then, the set of possible windows that can be represented with these values of p_k is

$$w_k(\tau) = \frac{(1-p_k)^2}{4}\left[\frac{A(\tau)}{B(\tau)}\right]^2 = \frac{(1-p_k)^2(1+\cos\tau)}{2(1+p_k^2-2p_k\cos\tau)}, \tag{2.7}$$

where $A(\tau)$ and $B(\tau)$ are the distances to the zero and the pole, as shown in Figure

2.1, and $g_k = (1 - p_k)^2/4$ is a normalization factor[2] to have 0 dB gain at time $\tau = 0$, that is, $w_k(0) = 1$.

While this filter is linear and time invariant (in fact frequency invariant[3]) a different time window is desired for each frequency component. Computing the response of the whole bank of filters for the entire spectrum sequence and then choosing the response for only one bin is computationally inefficient. For this reason, a Linear Time Variant (LTV) system, that consists in a Time–Varying (TV) IIR filter, is proposed as a way to approximate the filterbank response at the frequency bins of interest. It will no longer be possible to define the filter impulse response, as this could only be done if the filters were invariant to frequency shifts.

2.2.2 LTV IIR System

Selecting a different filter response of the filterbank for each frequency bin can be considered as applying an LTV system to the DFT of a frame. The desired response of the LTV for a given frequency bin is the impulse response of the correspondent filter.

Any LTV system can be expressed in the matrix form, $Y = K.X$ where K is the linear transformation matrix (also referred as Greens matrix) and, in this case, X is the DFT of the signal frame. A straightforward way to construct K for any LTV system is to set its i^{th} column as the response to a shifted delta $\delta[n - i]$, which is named Steady State Response (SSR).

The approach followed in this work consists in approximating the LTV system by a single TV IIR filter, assuming that the LTV system has a slow time–varying behavior and that its SSR can be implemented by an IIR filterbank. Then it is verified that the approximation is sufficiently good for our purposes. In the case of variable windowing to obtain a constant Q, these assumptions hold, as time windows for two consecutive frequency bins are intended to be very similar, and the LTV system can be implemented by an IIR filterbank as seen before.

[2]This normalization factor can be calculated from the impulse response evaluated at $n = 0$, or by the integral of the time window function.

[3]Note that we will use the usual time domain filtering terminology in spite of the fact that filtering is performed in the frequency domain.

A direct way of approximating the IIR filterbank is by a first order IIR of the form of equation 2.6, but in which the pole varies with frequency ($p = p[n]$),

$$Y[n] = X[n] + X[n-1] + p[n]Y[n-1]. \tag{2.8}$$

With an appropriate design, it reasonably matches the desired LTV IIR filterbank response, and its implementation has low computational complexity.

2.2.3 Time–Varying IIR filter design

A question that arises is how to design the TV IIR filter in order to have a close response to that of the LTV IIR filterbank. Several design criteria have been proposed in the literature [27], that may depend on the problem itself.

The TV IIR can also be represented by a matrix K_v in a similar way as the LTV filterbank, so the design can be done as in [27], by minimizing the normalized mean squared error, $E = ||K - K_v||^2/||K||^2$. In this work, the adopted design criteria is to impose the window behavior in time to obtain the desired constant Q. Then, the error is regarded as the difference between the desired Q and the effective obtained value. It becomes necessary to define an objective measure of Q. Usually the quality factor of a passband filter is defined as the ratio between the center frequency and the bandwidth at 3 dB gain drop. In our case the filtering is done in the frequency domain, so it is reasonable to measure Q in the time domain. Given that Q represents the number of cycles of an analyzed frequency component in the frame, it makes sense to define Q as the number of cycles within the window width at a certain gain drop, for example 3 dB. If τ'_k is the time at this drop for frequency f_k, $w_k(\tau'_k) = 10^{-\frac{3}{20}} w(0) \triangleq w'_k$, then $\tau'_k = Q/(2f_k)$. This definition allows the comparison of Q for methods with different window shapes. Note, however, that a similar measure of Q can be formulated in the frequency domain.

In the proposed approach the first step is to design an IIR filterbank that accomplishes the constant Q behavior. Then, a TV IIR filter is devised based on the poles of the filterbank. Finally a fine tuning is performed to improve the steadiness of the Q value for the TV IIR filter. In the following section, this procedure is described in detail.

2.2.3.1 Proposed design

Following the definition of Q in time, the poles of the IIR filterbank can be calculated from equation 2.7 as the solution of a second order polynomial:

$$(2w'_k - \cos(\tau'_k) - 1)p_k^2 + (2 + 2\cos(\tau'_k) - 4w'_k \cos(\tau'_k))p_k + 2w'_k - \cos(\tau'_k) - 1 = 0.$$

Then, a simple and effective design of the TV IIR filter consists in choosing for each frequency bin the corresponding pole of the IIR filterbank, that is $p[n] = p_k$, with $k = n$. The Q factors obtained with this approach are close to the desired constant value but with a slight linear drift. This result shows that the slow variation of the LTV system allows an approximation by a single TV IIR with a little deviation that can be easily compensated by adding the same slope to the desired Q value at each bin. Figure 2.4 shows the Q curve for the original and compensated designs, in which it can be seen that the actual obtained Q factor is almost constatnt along most frequencies. Only in the low frequency and high frequency ranges the quality factor deviates from the desired value. This happens as the frame length time limits the window analysis width for low frequencies, and the sampling frequency limits the representation of narrow windows necesary at high frequencies.

Another design consideration is that for low frequencies a constant Q would imply a longer window support than the frame time. It becomes necessary to limit the time τ'_k to a maximum time τ_{max}, such that 2 τ_{max} is smaller than the frame time. This limitation of τ'_k to a maximum value must be done in a smooth way. Let $\bar{\tau}'_k$ be a new variable that represents the result of saturating τ'_k. The transition can be implemented with a hyperbola whose asymptotes are $\bar{\tau}'_k = \tau'_k$ and $\bar{\tau}'_k = \tau_{max}$, so that $(\bar{\tau}'_k - \tau_{max})(\bar{\tau}'_k - \tau'_k) = \delta$, where δ is a constant that determines the smoothness of the transition.

The selection of τ_{max}, affects the behavior of the transform in low frequencies. Choosing a small τ_{max} compared to the frame time gives poor frequency resolution. On the contrary, if τ_{max} is set to a value close to the frame time, a better resolution is expected, but some distortion appears. This is because the time windows get close to a rectangular window for low frequencies. The spectrum of these windows has big side lobes, introducing Gibbs oscillations in the representation. Additionally, as a time window for low frequency approaches to a rectangular shape, its response to an impulse vanishes more slowly, so it becomes necessary to calculate the response for some negative frequency bins, adding extra

```
p = design_poles(NFFT,Q);
X = fft(fftshift(s));
Y'(1) = X(1);
for i = 2:NFFT/2
    Y'(i) = X(i-1) + X(i) + p(i)Y'(i-1);
end
Y(NFFT/2) = Y'(NFFT/2);
for i = NFFT/2-1:-1:1
    Y(i) = Y'(i+1) + Y'(i) + p(i)Y(i+1);
end
```

TABLE 2.1: Pseudocode of the TV IIR filter. First, the poles and normalization factor are designed given the number of bins (`NFFT`) and the Q value. Then the FFT of the signal frame `s` is computed after centering the signal at time 0. Finally the forward-backward TV IIR filtering is performed for that frame.

complexity. In practice it is reasonable to choose an intermediate value of τ_{max}, e.g. $\tau_{max} \approx 0.7\pi$, such that only for very low frequencies the transform exhibits non constant Q. Figure 2.2 shows details of the described poles design.

2.2.3.2 TV IIR filtering and zero-padding in time

It is common practice to work with a higher sampling frequency of the spectrum, typically obtained by zero-padding in time. In this case the TV IIR filter design changes, as the signal support becomes $(-\tau_1, \tau_1]$ with $0 < \tau_1 < \pi$. Then, the discontinuity to be avoided at the ends of the frame appears at $\pm\tau_1$, so a couple of zeros at $\pm\tau_1$ have to be placed instead of the zero at π. Window properties outside this support are irrelevant, as windowed data values are zero. The design of poles has to take into account the new zeroes and the time re-scaling, but windows with similar properties are obtained.

2.2.3.3 Implementation

The method implementation[4] is rather simple, as can be seen in the pseudocode of Table 2.1. A function to design the poles is called only once and then the forward-backward TV IIR filtering is applied to the DFT of each signal frame. The proposed IIR filtering applies a window centered at time 0, so the signal frame has to be centered before the transform. To avoid transients at the ends, the filtering should be done circularly using a few extra values of the spectrum as prefix and postfix. Their lengths can be chosen so as truncation error lies below a certain threshold, for instance 60 dB.

[4]The complete code is available at http://iie.fing.edu.uy/~pcancela/iir-cqt.

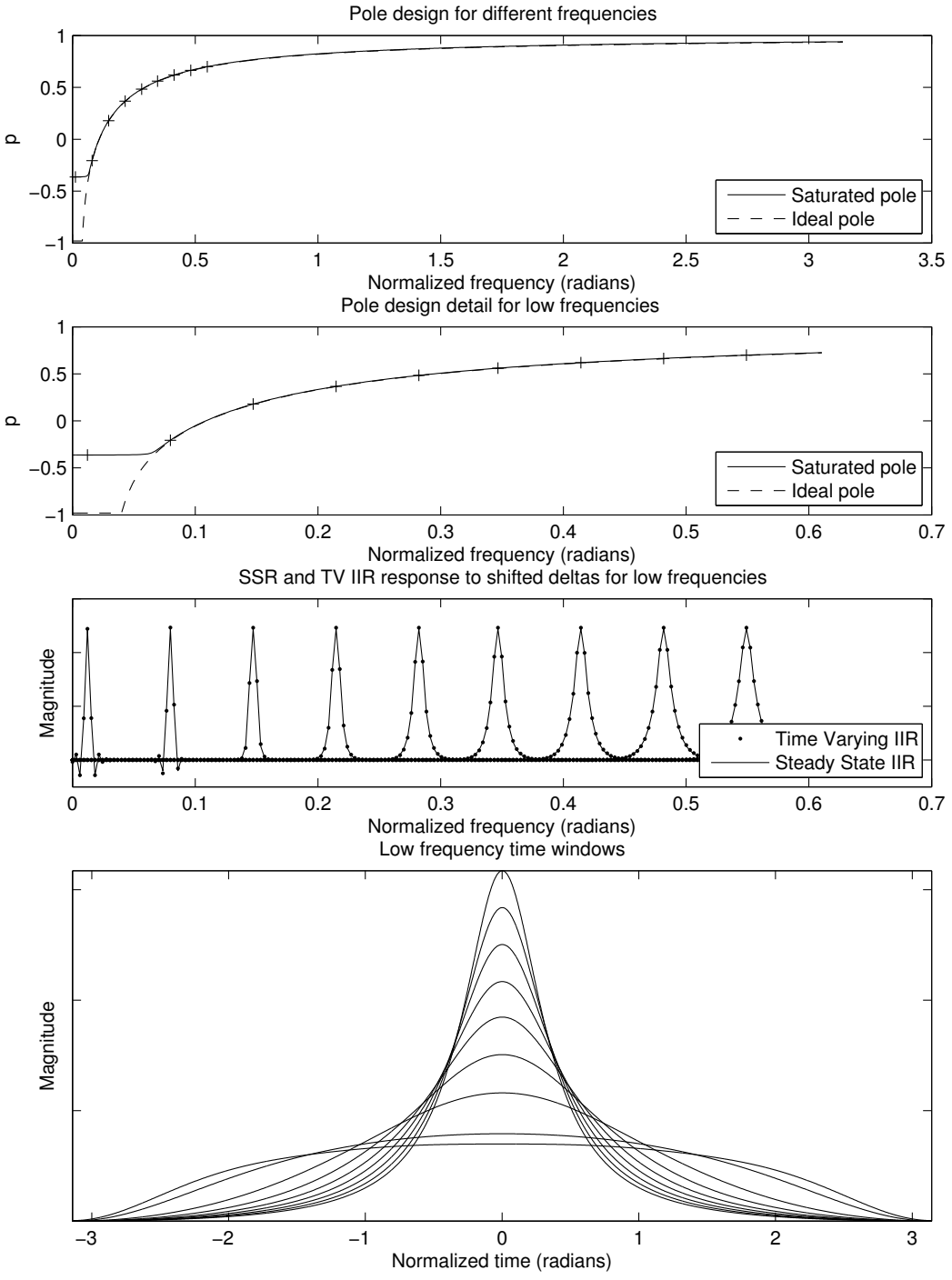


FIGURE 2.2: Detail of poles design. Pole locations for the ideal and saturated design. Impulse responses at low frequencies for the TV IIR and the Steady State, along with corresponding TV IIR time windows.

2.3 Methods Comparison

2.3.1 Frequency scale

Depending on the context of the music analysis application different frequency grids may be preferred. To this respect, the efficient CQT method can be designed for any arbitrary frequency spacing. On the contrary, the MR FFT and the IIR CQT are constrained to a linear frequency scale because they rely on the DFT. This spacing typically implies an oversampling at high frequencies to conform with the minimum spacing at low frequencies.

2.3.2 Effective quality factor

The analyzed methods have different flexibility to define an arbitrary Q at each frequency. The efficient CQT offers the freedom to set any possible Q for every bin. The MR FFT allows choosing the resolution for every bin from a reduced set not enabling an arbitrary Q. On the other hand, the TV IIR filter allows any Q value for any frequency but with the constraint that it evolves slowly with frequency. This holds particularly well in the case of a constant Q transform, so the IIR CQT can give any constant Q with a fairly simple design.

Figure 2.4 shows the obtained Q with the different methods. It can be observed that the MR FFT has a bounded Q due to the resolution quantization.

2.3.3 Windows properties

The spectral and temporal characteristics of windows at three different frequencies are shown in Figure 2.3 for each method. At frequency f_1, IIR CQT time window behaves like a Hann window. For lower frequencies it exhibits a flatter shape to extend the range of constant Q (see Figure 2.2). For higher frequencies, the main lobe of the obtained windows has a steeper drop up to -50 dB compared to a conventional Hann or Hamming window. As a counterpart, time resolution is slightly diminished. Note that the selected drop value in the definition of Q sets the location in this compromise.

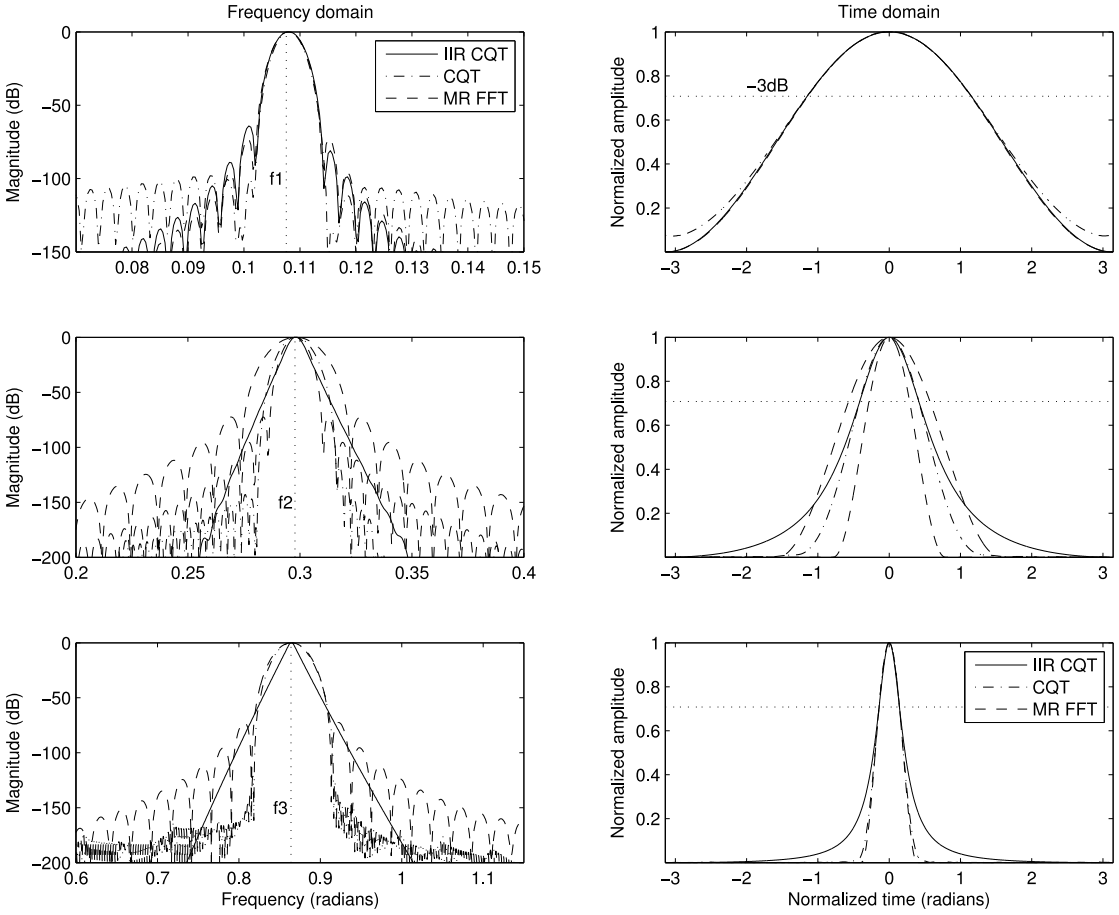


FIGURE 2.3: Windows comparison at frequencies f_1, f_2 and f_3 for the different methods. At f_1 and f_3 the three methods have the same Q, while at f_2 the MR FFT can not achieve the desired Q. For this reason, the two nearest MR FFT windows are considered at f_2. CQT and MR FFT are computed using a Hamming and Hann windows respectively.

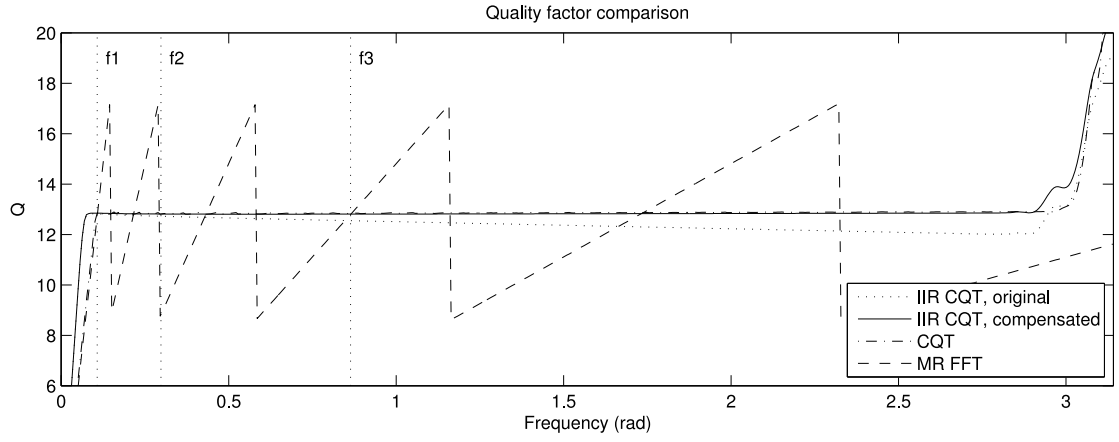


FIGURE 2.4: Comparison of the effective Q for a target value of 12.9 given the definition of 2.2.3. This value gives 34 cycles within the window, as commonly used in the CQT.

2.3.4 Computational complexity

The three algorithms are compared based on the number of real floating point operations performed in mean for each frequency bin. All of them compute the DFT of a non windowed frame, so these operations are not considered.

The number of operations in the efficient CQT depends on the length of the frequency kernels. This length varies with Q and is different for different frequency bins. For the Q and threshold values used in Figures 2.4 and 2.3 ($Q_{CQT} = 34$, $Q = 12.9$, $th = 0.0054$), `NFFT` $= 2048$ and $f_s = 44100$ Hz, the frequency kernel length varies from 1 to 57 coefficients, which implies a mean number of 27 real multiplications and 27 real additions. This result depends on the threshold and inversely on Q. The number of operations in this case is inversely proportional to the chosen quality factor. The MR FFT takes advantage of the hierarchical implementation of the FFT to compute the transform, so the windowing in the frequency domain needs only 3 complex sums and 2 multiplications for each bin. The total number of real floating point operations is then 4 multiplications and 6 additions per bin. The IIR CQT involves a forward and backward IIR filtering with a variable real pole and a zero, followed by a real normalization (see Table 2.1 for a pseudocode). As the frequency components are complex values, the necessary number of real operations to compute each bin is 6 multiplications and 8 additions (plus a negligible number of extra operations due to the circularly filtering approximation)per bin.

2.4 Examples and remarks

Finally, two different examples of the spectral analysis of polyphonic music using the proposed IIR CQT method are shown in Figure 2.5 together with conventional spectrograms. As it is expected in a constant Q transform, it can be noticed that singing voice partials with high frequency slope tend to blur in the spectrogram but are sharper in the IIR CQT. This improved time resolution in high frequencies also contributes to define more precisely the note onsets, as can be seen in the second example (e.g. the bass note at the beginning). Moreover, in the low frequency band, where there is a higher density of components, the IIR CQT achieves a better discrimination, due to the fact that its time windows are flatter than typically used windows. At the same time, frequency resolution for the higher partials of notes with a steady pitch is deteriorated.

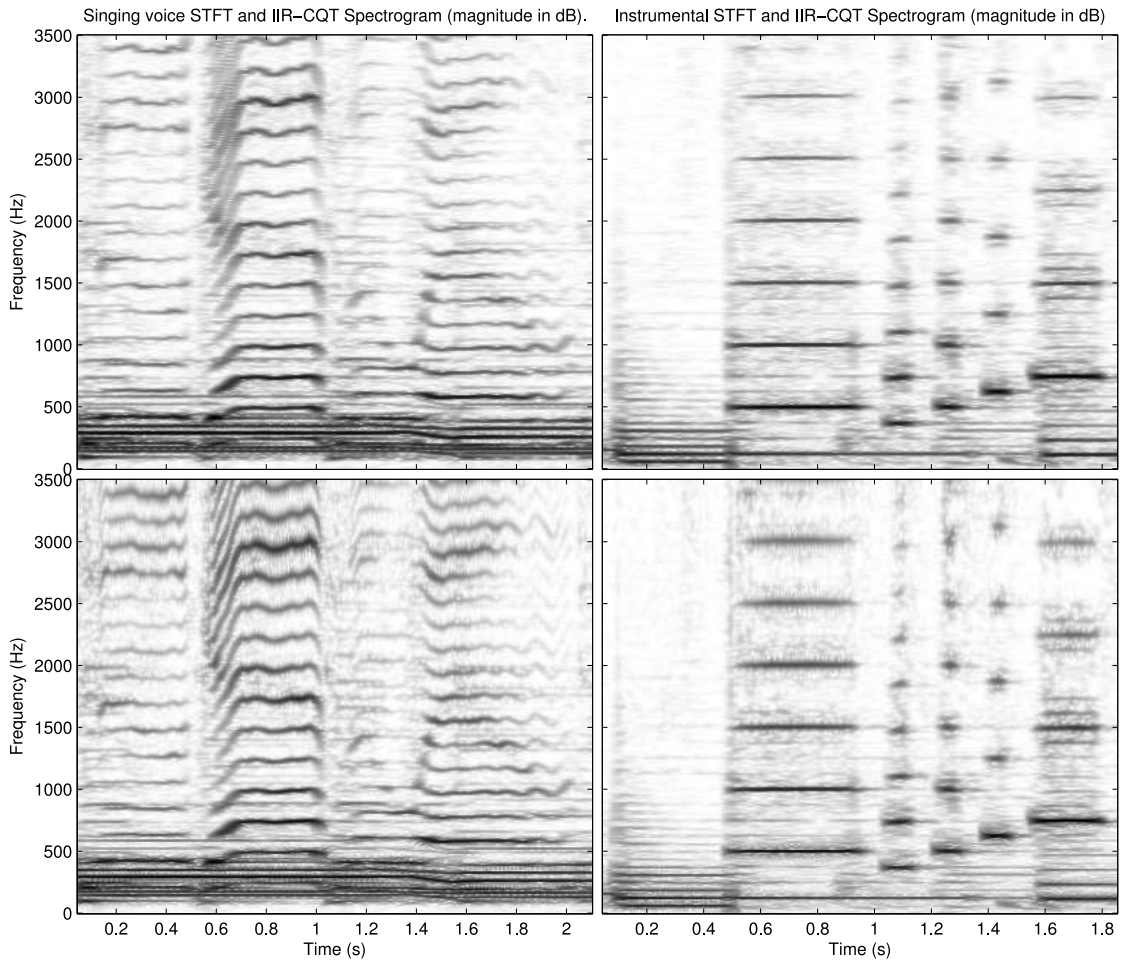


FIGURE 2.5: STFT and IIR CQT for two audio excerpts, one with a leading singing voice and the other, instrumental music.

A novel method for computing a constant Q spectral transform was proposed and compared with two existing techniques. It shows to be a good compromise between the flexibility of the efficient CQT and the low computational cost of the MR FFT. Results show that it seems to be a good spectral representation tool for audio signal analysis algorithms.

Chapter 3

Fan Chirp Transform

The Fan Chirp Transform (FChT), which was designed to achieve a better representation of non-stationary harmonic signals, was originally proposed in [13] to represent speech signals. As it was described in the Introduction, a wide set of sounds have a harmonic structure that slowly changes in time, which the FChT can represent with better resolution than the STFT. A map called F0gram is calculated based on the FChT to represent harmonic content in a musical excerpt.

In this Chapter we describe an independent development of the FChT proposed by the author oriented to represent harmonic sources in polyphonic music. The differences arise in criteria to choose the optimum representation window used in the transform and also to attenuate spurious peaks that appear as a consequence of representing multiple sources. This Chapter is based on the publication [20], developed in collaboration with Martín Rocamora and Ernesto López. The description reproduces some of the article passages, it includes also some modifications or additions in order to contribute to the structure of this document.

3.1 Fan Chirp Transform

3.1.1 Formulation

In this work, the definition of the FChT adopted is,

$$X(f,\alpha) \triangleq \int_{-\infty}^{\infty} x(t)\ \phi'_{\alpha}(t)\ e^{-j2\pi f \phi_{\alpha}(t)} dt, \tag{3.1}$$

where $\phi_\alpha(t) = (1 + \frac{1}{2}\alpha t)\, t$, is a time warping function. This was formulated independently from the original work [13], so the properties are slightly different as will be indicated later. Notice that by the variable change $\tau = \phi_\alpha(t)$, the formulation becomes,

$$X(f, \alpha) = \int_{-\infty}^{\infty} x(\phi_\alpha^{-1}(\tau))\; e^{-j2\pi f \tau} d\tau, \tag{3.2}$$

which can be regarded as the Fourier Transform of a time warped version of the signal $x(t)$, and enables an efficient implementation based on the FFT. The goal pursued is to obtain an acute representation of linear chirp signals of the form $x_c(t, f) = e^{j2\pi f \phi_\alpha(t)}$. These chirps are chosen because they consider a first order approximiation of the pitch pitch contours. Considering a limited analysis time support, the analysis basis is $\Gamma = \{\gamma_k\}_{k\in\mathbb{Z}}$, $\gamma_k = \phi'_\alpha(t)\; e^{j2\pi \frac{k}{\breve{T}} \phi_\alpha(t)}$, $t \in \left[\phi_\alpha^{-1}(-\frac{\breve{T}}{2}), \phi_\alpha^{-1}(\frac{\breve{T}}{2})\right]$. The inner product of the chirp and a basis element results in,

$$\begin{aligned}
\left\langle x_{ch}(t, 2\pi \tfrac{l}{\breve{T}}), \gamma_k \right\rangle &= \frac{1}{\breve{T}} \int_{\phi_\alpha^{-1}(-\frac{\breve{T}}{2})}^{\phi_\alpha^{-1}(\frac{\breve{T}}{2})} \phi'_\alpha(t)\; e^{j2\pi \frac{l-k}{\breve{T}} \phi_\alpha(t)} dt \\
&= \frac{1}{\breve{T}} \int_{-\frac{\breve{T}}{2}}^{\frac{\breve{T}}{2}} e^{j2\pi \frac{l-k}{\breve{T}} \tau} d\tau \qquad (3.3) \\
&= \delta[l-k], \qquad (3.4)
\end{aligned}$$

which denotes that only one element of the basis represents the chirp. Note that the limits of integration include an integer number of cycles of the chirp, in the warped and the original time interval.

In [13] the basis are designed to be orthonormal, in order to obtain perfect reconstruction directly from the analysis basis. However, its response to a chirp of constant amplitude is not represented by a single element. It is important to note that when the signal is windowed the orthogonality disappears so as the perfect reconstruction. In a similar way, result given by equation 3.3 does not hold anymore. To that end, it is worth defining a more appropriate goal, that is what kind of response would be desirable for a time limited chirp. The solution proposed in this work permits to achieve a delta convolved with the Fourier Transform of a well-behaved analysis window. This motivates the above definition of the analysis basis Γ and the application of the analysis window to the time warped signal (which also differs from [13]). Then, the here proposed FChT for a time limited support is

$$X_w(f, \alpha) = \int_{-\infty}^{\infty} x(t)\; w(\phi_\alpha(t))\; \phi'_\alpha(t)\; e^{-j2\pi f \phi_\alpha(t)} dt, \tag{3.5}$$

where $w(t)$ stands for a time limited window, such as Hanning.

Consider the case of a signal composed of L harmonically related linear chirps, $x_{hc}(t, f_0, L) = \sum_{k=1}^{L} e^{j2\pi k f_0 \phi_\alpha}$. All components share the same fan chirp rate α, so applying the appropriate warping ϕ_α delivers constant frequency harmonically related sinusoidal components. The FChT representation therefore shows a sharp harmonic structure as it is composed of deltas convolved with the Fourier Transform of the window.

3.1.2 Discrete time implementation

As stated in the previous section, the FChT of a signal $x(t)$ can be computed by the Fourier Transform of the time warped signal $\breve{x}(t) = x\left(\phi_\alpha^{-1}(t)\right)$, where

$$\phi_\alpha^{-1}(t) = -\frac{1}{\alpha} + \frac{\sqrt{1+2\alpha t}}{\alpha}. \tag{3.6}$$

This warping function transforms linear chirps of instantaneous frequency $\nu(t) = (1+\alpha t) f$ into sinusoids of frequency $\breve{\nu}(t) = f$. In practice, the original signal is processed in short time frames. In order to properly represent it with its warped counterpart, temporal warping is implemented by adopting the following criteria. After the time warping, the frequency of the resulting sinusoid is the frequency of the linear chirp at the centre of the analysis window. Besides, the amplitude value of the warped signal remains unchanged in the central instant of the window. Note this implies that the duration of the original signal and the warped signal may be different and is not imposed in [13].

Let $x[n] = x((n-(N-1)/2)/f_s - t_a)$ be a finite length signal frame at central time t_a, where $n = 0, \ldots, N-1$. When working with discrete signals, the temporal warping is implemented by non-uniform resampling of $x[n]$. If the sampling frequency is f_s, the frame duration is $T = (N-1)/f_s$. The time instant corresponding to the $n-th$ sample of $x[n]$ is defined as $t_n = (n-(N-1)/2)/f_s$. Thus, the time domain of the signal is $\mathcal{D}_x = [-T/2, T/2]$. Similarly, let $\breve{x}_\alpha[m]$ be the time warped signal, with $m = 0, \ldots, M-1$, and sampling frequency $\breve{f}_s$. Its duration is then $\breve{T} = (M-1)/\breve{f}_s$ and the time instant of the m-th sample is defined as $\breve{t}_m = (m-(M-1)/2)/\breve{f}_s$. The warped time domain is $\mathcal{D}_{\breve{x}} = [-\breve{T}/2, \breve{T}/2]$. To compute the sample corresponding to time instant $\breve{t}_m$ of the warped signal, it is necessary to evaluate $x[n]$ at time instant $t_m = \phi_\alpha^{-1}(\breve{t}_m)$. As this instant may not

coincide with a sampling time, the evaluation must be done using some interpolation technique. Time warping process is illustrated in figure 3.1. The last step of the FChT is to apply an analysis window to the time warped signal and compute the DFT.

The transform parameters are the number of samples M and the sampling rate $\breve{f}_s$. They should be selected in order to avoid aliasing in the resampling process of the original signal $x[n]$. Given a sampling rate f_s, suppose that the signal is band limited to $f_{\max}$ (with $f_{\max} \leq f_s/2$). To set M and $\breve{f}_s$ to avoid aliasing, the maximum resampling period $T_s^{\max}$ must fulfill $T_s^{\max} \leq 1/(2f_{\max})$. If $\alpha \geq 0$, the previous condition is met if,

$$\left(\phi_\alpha^{-1}\right)' \left(-\frac{\breve{T}}{2}\right) \leq \frac{T_s^{\max}}{\breve{T}_s} = \frac{\breve{f}_s}{2f_{\max}},$$

where $\breve{T}_s = 1/\breve{f}_s$ is the sampling period of $\breve{x}_\alpha[m]$. Using equation 3.6 and considering that $\breve{T} = (M-1)/\breve{f}_s$, the condition becomes,

$$\breve{f}_s \geq \frac{2f_{\max}}{\sqrt{1 - |\alpha| \frac{M-1}{\breve{f}_s}}}.$$

The length N of the analysis window must be large enough to be able to interpolate $x[n]$ at every time instant t_m. Specifically, it must fulfill $T/2 \geq \max_m |t_m|$. If $\alpha \geq 0$, $\max_m |t_m| = |t_0| = |\phi_\alpha^{-1}(-\breve{T}/2)|$ and thus $T/2 \geq |\phi_\alpha^{-1}(-\breve{T}/2)|$, which leads to

$$N > 2f_s \frac{1 - \sqrt{1 - |\alpha| \frac{M-1}{\breve{f}_s}}}{|\alpha|}.$$

In the current implementation, the discrete signal, originally sampled at 44100 Hz, is first low-pass filtered to limit the spectral content up to $f_{\max} = 10000$ Hz, and then upsampled to double the sampling rate, so $f_s = 88200$ Hz. The computation of the warped samples is done using linear interpolation. The above mentioned upsampling is performed to obtain a more accurate and efficient interpolation. The maximum absolute value of the fan chirp rate employed is $\alpha_{\max} = 6$. With these parameters, values of $M = 2048$, $\breve{f}_s = 30000$ Hz and $N = 10000$ meet the above conditions. Note that the proposed implementation permits to choose M as a small power of two to take advantage of the FFT efficiency.

Another consideration regarding the implementation is that the time warping design is performed numerically based on relative instantaneous frequency functions.

More precisely, the design begins with the selection of the warping instantaneous frequency $f_r[n]$ for each sample. Then, the function $\phi[n]$ is obtained by numerical integration of $f_r[n]$. Finally the function $\phi^{-1}[n]$, needed to compute the resampling times, is obtained by numerical inversion. This allows the implementation of arbitrary warping functions instead of only linear warpings, which will be discussed in Section 4.

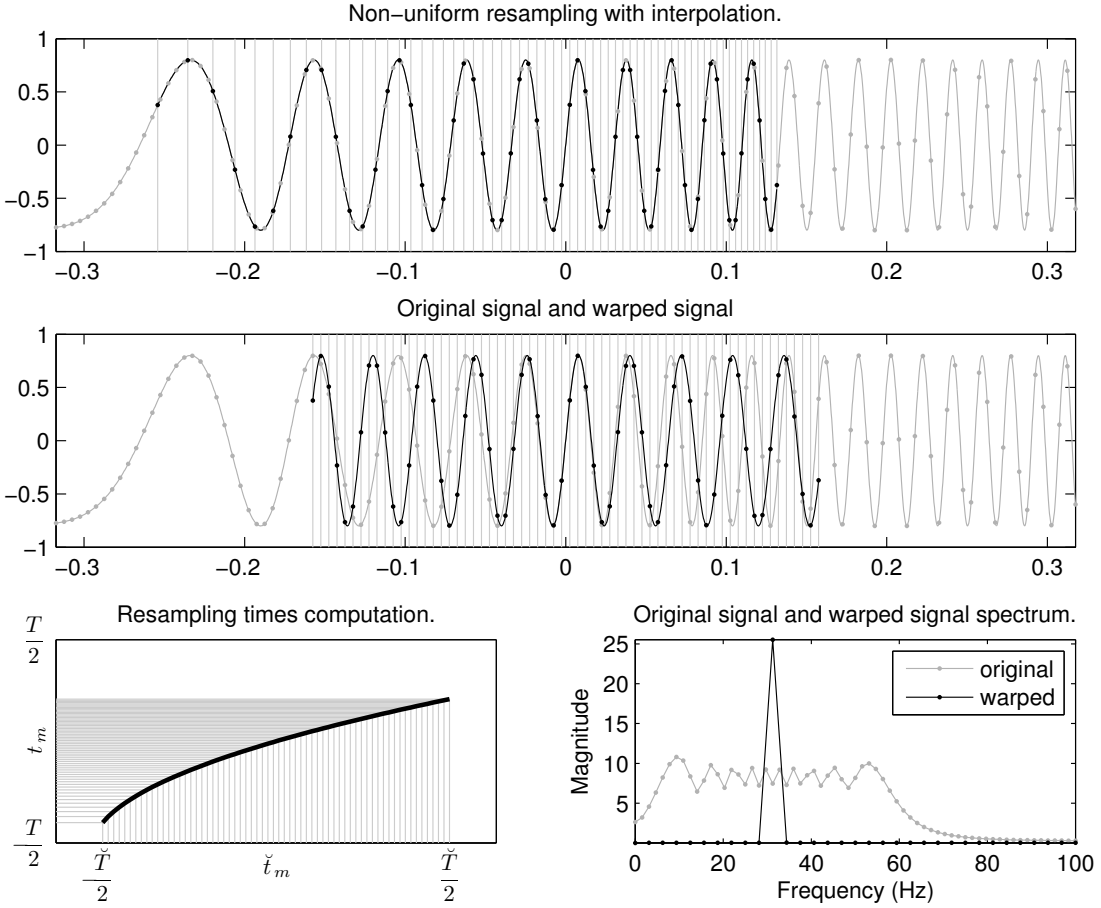


FIGURE 3.1: Warping process illustration. A sinusoid is obtained by appropriately warping a linear chirp. Note that the central time instant remains the same and the time supports are different. The FChT of the linear chirp shows a sharp high peak.

3.2 Fan Chirp Transform for music representation

In a practical situation, real signals such as speech or music sounds can be assimilated to harmonically related linear chirps only within short time intervals,

where the evolution of frequency components can be approximated by a first order model. This suggests the application of the FChT to consecutive short time signal frames, so as to build a time–frequency representation as a generalization of the spectrogram [13]. In the monophonic case, a single value of the fan chirp rate α that best matches the signal pitch variation rate should be determined for each frame. This is the key factor to obtain a detailed representation using the FChT. Different approaches could be followed, such as predicting the pitch evolution and estimating α as the relative derivative of the pitch [13].

In the polyphonic case, there is no single appropriate value of α, because the multiple harmonic sounds present are likely to change their fundamental frequency (f_0) differently within the analysis frame. For this reason, a multi-dimensional representation for each frame seems better suited in this case, consisting in several FChT instances with different α values. A given FChT is tuned to represent one of the harmonic sounds with reduced spectrum spread, whereas poorly representing the remaining ones. The selection of a reduced set of α values for each frame that produce the better representation of each existing sound can be tackled by means of sinusoidal modeling techniques as in [16]. In this work a straightforward exhaustive approach is adopted, that consists in computing a dense (f_0, α) plane and selecting the best chirp rates based on pitch salience. In addition, the pitch salience computation from the FChT produces itself a detailed representation of the melodic content of the signal, that can be useful in several applications. This is described in detail in the following section.

3.3 Pitch salience computation

The aim of pitch salience computation is to build a continuous function that gives a prominence value for each fundamental frequency in a certain range of interest. Ideally it shows pronounced peaks at the positions corresponding to the true pitches present in the signal frame. This detection function typically suffers from the presence of spurious peaks at multiples and submultiples of the true pitches, so some sort of refinement is required to reduce this ambiguity. A common approach for pitch salience calculation is to define a fundamental frequency grid, and compute for each frequency value a weighted sum of the partial amplitudes in a whitened spectrum. A method of this kind was used in [28] for melody extraction, which is formulated in the following according to the log-spectrum gathering proposed in [14].

3.3.1 Gathered log-spectrum (GlogS)

The salience of a given fundamental frequency candidate f_0 can be obtained by gathering the log-spectrum at the positions of the corresponding harmonics as in [14],

$$\rho_0(f_0) = \frac{1}{n_H} \sum_{i=1}^{n_H} log|S(if_0)|, \tag{3.7}$$

where $|S(f)|$ is the power spectrum and n_H is the number of harmonics that are supposed to lie within the analysis bandwidth. Linear interpolation from the discrete log-spectrum is applied to estimate the values at arbitrary frequency positions. The logarithm provides better results compared to the gathering of the linear spectrum. This makes sense, since the logarithm function can be regarded as a kind of whitening in order to make the pitch salience computation more robust against formant structure and noise. With this respect, it is interesting to note that a p-norm with $0 < p < 1$ is also appropriate and shows similar results. Note that this seems coherent with the use of more robust norms which is advocated in sparsity research. Therefore, the actual implementation is $log(\gamma|S(if_0)|+1)$ which adds the flexibility to custom the norm applied by means of the γ parameter[1].

3.3.2 Postprocessing of the gathered log-spectrum

The harmonic accumulation shows peaks not only at the position of the true pitches, but also at multiples and submultiples (see Figure 3.2). To handle the ambiguity produced by multiples, the following simple non-linear processing is proposed in [14],

$$\rho_1(f_0) = \rho_0(f_0) - \max_{q \in \mathbb{N}} \rho_0(f_0/q). \tag{3.8}$$

This is quite effective in removing pitch candidates multiples of the actual one (as can be seen in Figure 3.2). When dealing with monophonic signals this suppression is enough. If pitch estimation is obtained as the position of the maximum of $\rho_1(f_0)$, $\hat{f}_0 = \arg\max \rho_1(f_0)$, submultiple spurious peaks do not affect the estimation because their amplitude is necessarily lower than for the true pitch. However, in the polyphonic case, submultiple peaks should also be removed. For this reason, the detection function is further processed to remove the (k-1)-th submultiple according to,

$$\rho_2(f_0) = \rho_1(f_0) - a_k \, \rho_1(kf_0) \tag{3.9}$$

[1] Higher values tend to a 0-norm while lower values tend to a 1-norm. All the results reported correspond to $\gamma = 10$.

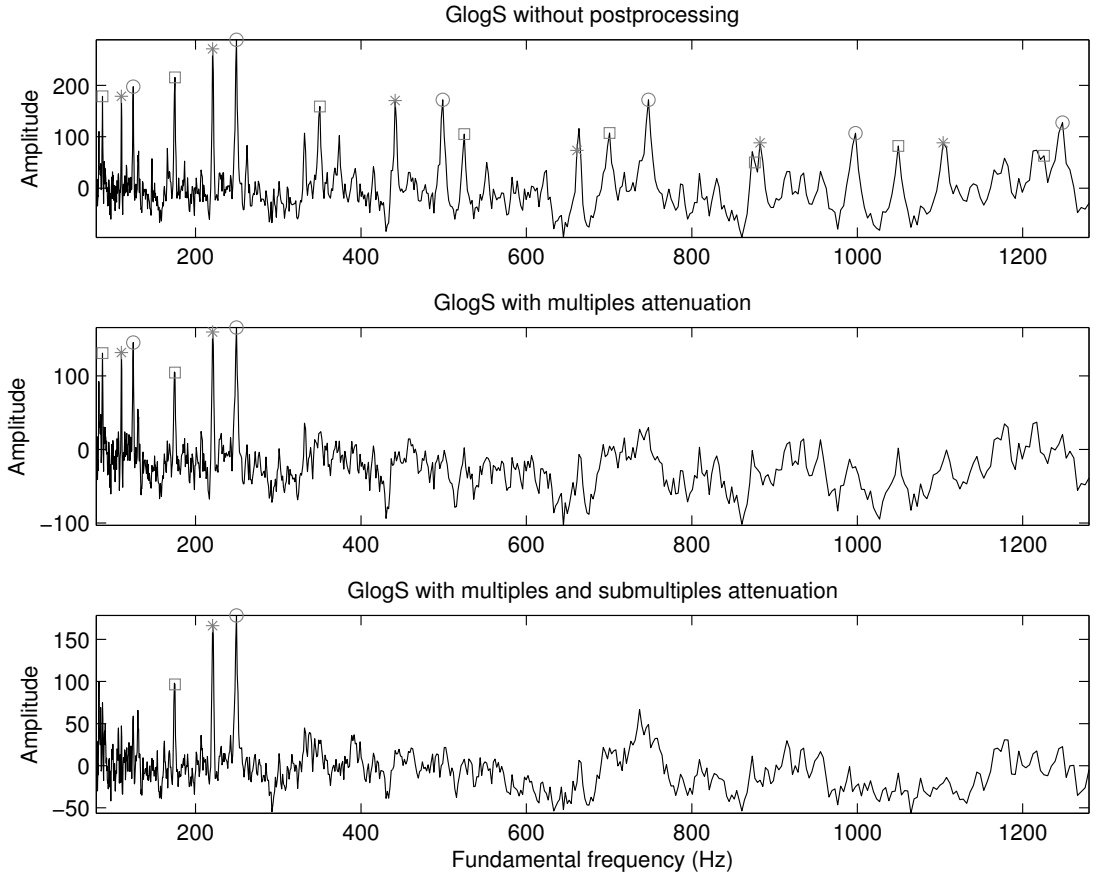


FIGURE 3.2: Normalized gathered log spectrum and the postprocessing stages for a frame of the audio excerpt at $t = 0.36s$, with three prominent simultaneous singing voices and low accompaniment. Positions of each corresponding f_0, multiples and first submultiple are also depicted.

where a is an attenuation factor. From the simulations conducted it turned out that removing only the first submultiple ($k = 2$) is commonly sufficient for melodic content visualization and melody detection (see Figure 3.2). For a single ideal harmonic sound, following a similar reasoning to that of [14], it can be shown that the attenuation factor is $a_2 = 1/2$. However, it can also be shown that the variance of $\rho_0(f_0)$ is proportional to the fundamental frequency (see also [14] appendix B). In practice a true pitch peak can be unnecessarily attenuated due to the large variance at its multiple, so a more conservative attenuation factor is preferred. Slightly better results were obtained over polyphonic music for $a_2 = 1/3$, and this is the value used for the reported results.

3.3.3 Normalization of the gathered log-spectrum

The variance increase with f_0 is an undesired feature. When applied to melodic content visualization different frequency regions are unbalanced and it leads to incorrect detections when pursuing melody extraction. For this reason, the last step in pitch salience computation is to normalize $\rho_2(f_0)$ to zero mean and unit variance. To do this, the mean and the variance of $\rho_2(f_0)$ are collected at each f_0 for every frame from a music collection (the complete RWC Popular Music Database [29] was used for this purpose). Each one of these statistics are then approximated by a second order polynomial, as illustrated in Figure 3.3. The polynomials evaluated at each f_0 are the model used to obtain a normalized gathered log-spectrum $\overline{\rho}_2(f_0)$. The fundamental frequency grid used is logarithmically spaced with 192 points per octave.

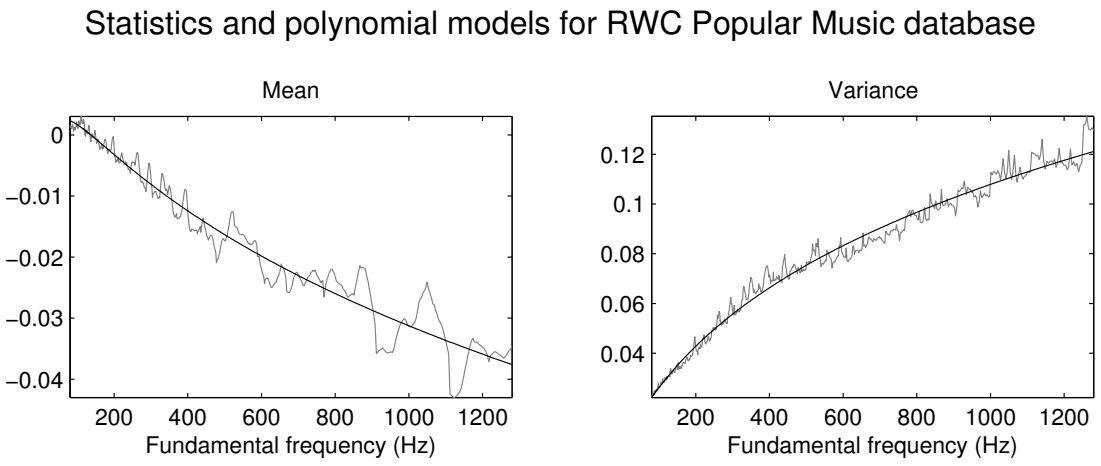


FIGURE 3.3: Gathered log spectrum normalization model.

3.3.4 Fan chirp rate selection using pitch salience

As early mentioned, the α values that best represent the different harmonic sounds in a signal frame are selected by means of pitch salience. Several FChT instances are computed for each frame using different α values. For each FChT a gathered log spectrum is calculated as described above, so as to build a dense pitch salience plane $\overline{\rho}_2(f_0, \alpha)$. See Figure 3.4 for an example of this dense pitch salience plane. Given a sound source of fundamental frequency $\hat{f}_0$, the energy of its harmonics is more concentrated at the FChT instance corresponding to the best matching α value $\hat{\alpha}$. Therefore, the value of $\overline{\rho}_2(\hat{f}_0, \hat{\alpha})$ is the highest among the different available α values. For this reason, a different α value is selected for each f_0 in the

grid, giving a single pitch salience value for each f_0 (see Figure 3.4). The reduced set of α values can be selected according to their corresponding pitch salience.

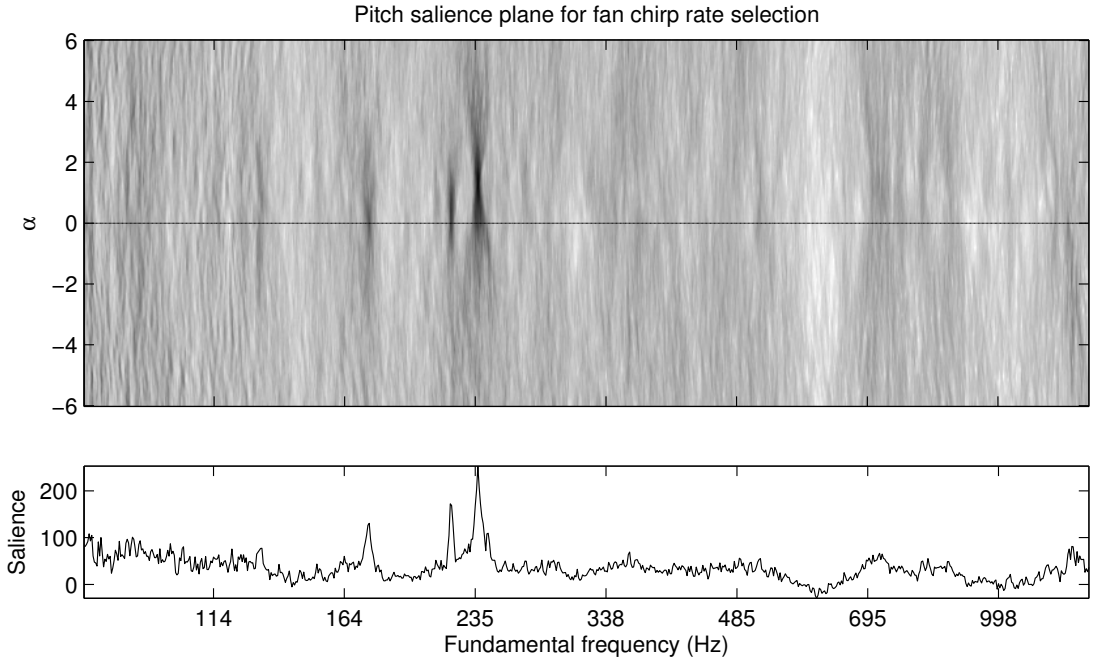
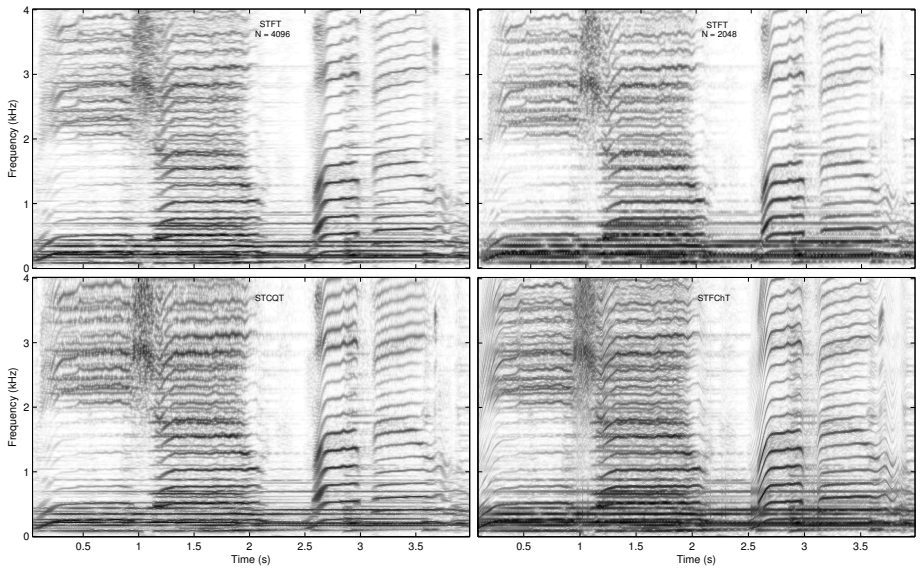


FIGURE 3.4: Pitch salience plane $\overline{\rho}_2(f_0, \alpha)$ for a frame of the audio excerpt at $t = 0.27s$. Prominent salience peaks (darker regions) can be distinguished corresponding to the three singing voices. Note that two of them are located approximately at $\alpha = 0$ and one at $\alpha = 1.3$. This indicates that two of the voices are quite stationary within the frame while the other is increasing its pitch. The maximum pitch salience value for each f_0 is also depicted.

Figure 3.5 shows a comparison of time–frequency representations which includes the FChT using the α that gives the most prominen pitch salience at each frame which was calculated with the proposed method. Figure 3.6 shows a comparison of the same transforms for a single frame at a time in which high pitch rate change. There is a clear resolution improvement in this frame when the FChT is claculated wth the correct α.

3.4 Conclusions

An independent formulation of a non classical time–frequency representation, the FChT, was developed for polyphonic music analysis. The formulation presented provides an acute representation of harmonically related linear chirp signals. The



)

FIGURE 3.5: Time–Frequency representations comparison for an audio excerpt of the music file "pop1.wav" from the MIREX [22] melody extraction test set. It consist of three simultaneous prominent singing voices in the first part followed by a single voice in the second part, and a rather soft accompaniment whithout percussion. The representations depicted are: Spectrograms for window length of 4096 and 2048 samples at $f_s = 44100$ Hz, a Short Time CQT and a Short Time FChT using the herein proposed method. Note the improved time–frequency resolution for the most prominent singing voice in the latter representation, especially in the time at which there is a fast pitch change.

implementation introduced was devised to be computationally manageable and enables the use of non linear warpings. Both the formulation and the implementation differ from early proposals [13]. In order to precisely represent the different pitched sources in a signal using the FChT an existing method based on pitch salience was adopted, which was improved and adapted to handle polyphonic music.

Different Music Information Retrieval applications can be approached with algorithms based in the FChT as a low level representation. In Section 5 we describe some techniques based on the FChT applied to solve different MIR problems and applications. Further information can be found in the related published articles describing these works. These, include:

- The pitch salience computation from the FChT can be used as a melodic content visualization tool. Results obtained for a frame based melody detection evaluation shows that the introduced F0gram can be used as a front-end for music analysis.

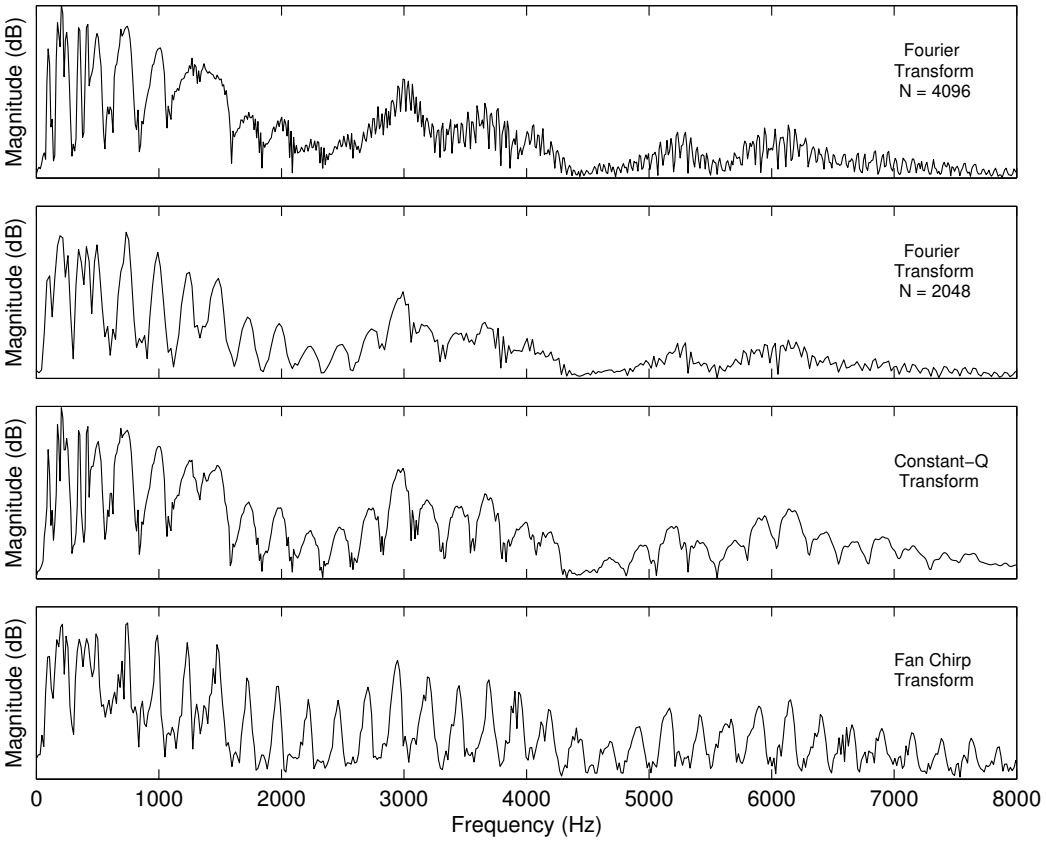


FIGURE 3.6: Time–Frequency representations comparison for a frame of the audio excerpt at time instant $t = 2.66s$. The prominent singing voice has a very high pitch change rate at this instant. This produces a blurry representation of the strongly non-stationary higher harmonics in the Fourier Transform. The representation of these harmonics is improved with the CQT because of the use of shorter time windows in high frequency. The FChT exhibits a more clear harmonic peak structure for the fan chirp rate that best represents the pitch change rate of the singing voice.

- A similar method to the herein described (adding temporal tracking) was submitted to MIREX 2008 Audio Melody Extraction Contest [28], performing best on Overall Accuracy.

Chapter 4

Fan Chirp Transform Analysis and Extensions

In this Chapter we will analyze further the time–frequency representation properties of the FChT, and make a comparison with classical techniques such as the STFT. We also discuss a series of different modifications and extensions that can be applied to the Fan Chirp Transform. The analysis of these extensions is performed to show the potential and the limitations of the method.

4.1 Time–Frequency Analysis Windows

Most time–frequency representations are based on the use of time-defined analysis windows. The characteristics of these windows determine the properties and impact the quality of the representation. The classical STFT spectrogram can be taken as a reference for comparison. The most relevant characteristic of the window is its length, which restricts the frequency resolution in the time–frequency resolution trade-off. Another key design factor is the choice of the shape of the window. Different criteria is usually set in the frequency domain concerning side-lobes rejection. Some of the typical windows used in literature are Hanning, Hamming, Blackman-Nuttall, among others.

The Hanning window is based on a raised cosine, having good continuity properties in the ends, and the secondary lobes drop at about 18dB per octave. The Hamming window is optimized to minimize the first side lobe, having a height of about a fifth of the Hanning one. Blackman windows use higher order cosine terms to

optimize different properties; as a general rule, they produce a slightly wider main lobe, but have an excellent side lobes rejection, with different decays and maxima dependent on the chosen criteria.

Figure 4.1 shows some classical windows time and frequency characteristics, which guide their selection in different applications. It is also common practice to study their behavior in the time–frecuency plane, but regarding a specific signal to be studied. For example to see how a sinusoid is represented or a sinusoid with a short attack time, so that the trade-off between time and frequency can be observed.

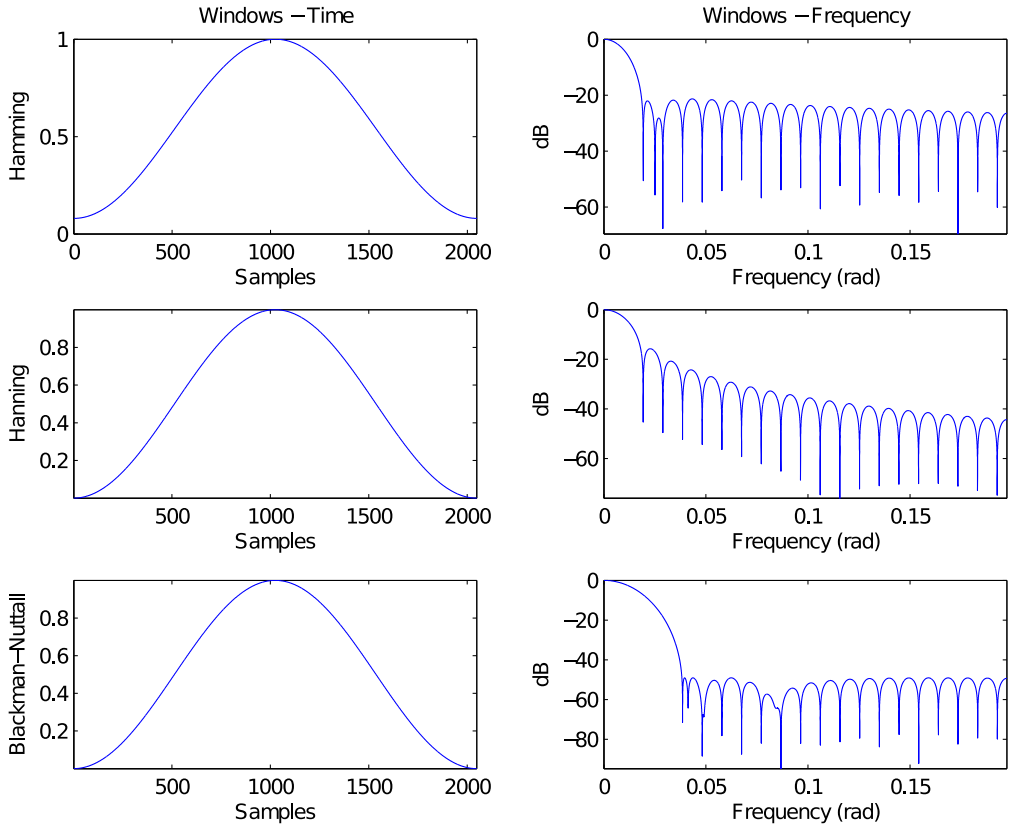


FIGURE 4.1: Typical analysis windows properties in time and frequency domain.

The definition of the linear Fan-Chirp Transform adds a new parameter, namely the chirp rate α. As this new parameter can take different values in a continuous range, it can be interpreted as a new dimension of the representation space. In the case of linear chirps, the new dimension of analysis α corresponds to the slope of the relative frequency change which marginalizes the energy in a direction with a slope proportional to the central frequency value. The design criteria of the analysis should not be restricted to the characteristics exhibited for $\alpha = 0$, but include this new dimension range of values α.

The FChT allows the use of longer windows with the correspondent frequency resolution gain when the appropriate α is selected. This happens because the energy is marginalized along a direction that is parallel to the partials, making loss of resolution be aligned to them, with a low impact in the overall resolution trade-off as the amplitude of the partials changes slowly in time. The possibility to extend the window analysis time with the FChT representation permits to accurately describe a signal under certain hypothesis which are fulfilled when the signal has a harmonic structure. In addition, as a consequence of using longer windows in time, some artifacts become more noticeable, and can degrade the representation. Particularly, a constructive and destructive interference pattern appears when describing a harmonic signal with a significant pitch variation. In the next section we will analyze why these patterns appear, what are the consequences in the representation and how they can be attenuated.

4.1.1 Analysis Windows in the original time domain

After the appropriate time warping is applied, all the harmonics of a source would be converted into sinusoids for a given time. The analysis window is designed to give the best possible resolution in the warped time domain, in other words the best to represent the obtained sinusoids. As a consequence, the analysis window in the original time domain does not fulfill the typical desirable properties of an analysis window. Specifically, the asymmetry of the windows can have a sensible impact in the quality of the FChT of the F0gram as a time–frequency representation. For big chirp rates, the windows in the original time domain are fairly asymmetric, Figure 4.2 shows typical shapes for different values of α. The shape of the window itself does not play an important role in the original time domain, but affects the criteria of the time placement of the window. For the selected criteria, the center of the window in the time warped domain is mapped to coincide with the analysis time in the original time domain. This makes the peak of the window to be placed at the reference analysis time. While this criteria works reasonably good, the windows for big positive and negative values shift their center of mass to previous and future times, giving results in the representation that correspond conceptually to different times.

A more appropiate window centering criteria would be to use the center of mass or center of energy of the window. Given a time warping function $\phi(t)$, and an

analysis window $W(t)$, we can define the center of mass as,

$$t_m = \int_{-\infty}^{\infty} t|W(\phi(t))|dt$$

and the center of energy as:

$$t_e = \int_{-\infty}^{\infty} tW^2(\phi(t))dt$$

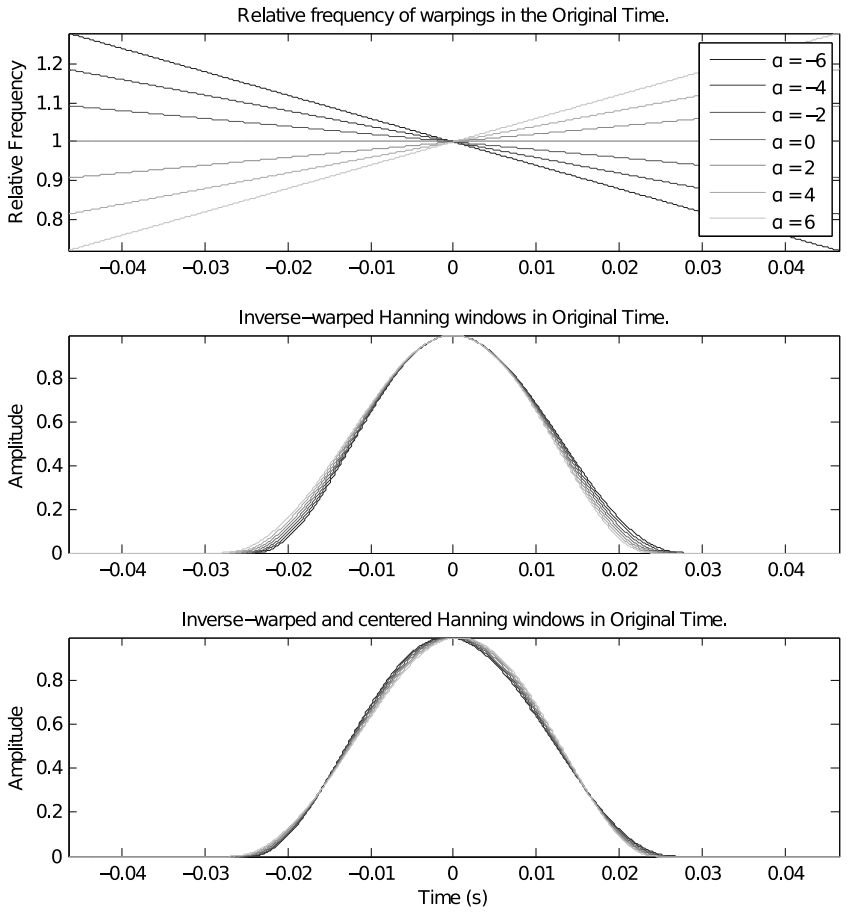


FIGURE 4.2: Example of different chirp-rate relative frequencies mapping and the corresponding inverse-warped Hanning windows in the original time domain. In the bottom, the same windows are unwarped after shifting the Hanning windows in the warped time to have their center of mass aligned.

So that the windows in the original time domain should be $W(\phi(t) - t_m)$ and $W(\phi(t) - t_e)$ respectively. Figure 4.2 shows the windows in the original time domain, and how they are shifted in time when warped. It also shows the same windows with their center of mass aligned. Using this new criteria, the set of transforms for different chirp-rates becomes more consistent, as they describe or

represent more accurately the properties of the spectrum for the same time reference. If this correction is not done, the time analysis is slightly shifted to the future or past depending on the chirp rate of analysis. This would make inadequate the comparison of the selection of the best chirp rate at a certain time slot as the comparison is based in the analyis of the signal in slightly different times.

The practical implementation does not require extra processing in the case of classical windows, as for each chirp rate, the correspondent time shift in the warped time domain can be precalculated to achieve the desired correction shifts. The analysis window in the warped time is then applied with the necessary shift prior to the Fourier Transform. In the case of the constant Q windowing, it is not as simple, but can also be easily corrected. The time shift will be different for every window corresponding to each frequency, thus the first step is to calculate a shift function of frequency $t_{sh}(f)$. As the shift depends on the frequency, the correction cannot be done in the time domain, as the same shift would be applied to all the windows. In this case the shift has to be performed in the frequency domain. If we multiply the spectrum of the signal by a function $F_{sh}(f) = e^{-jft_{sh}(f)}$, a group delay $t_{sh}(f)$ will be applied at frequency f components, obtaining the desired time shift.

The Fan-Chirp Transform proposed in [13] was designed to set the analysis window in the original time domain, while this selection does not bring optimal representation of sinusoids in the warped time, it offers a good centering criteria as they are symmetric and then do not need any time correction. We believe that both defining the windows in the warped time with the appropriate time shift correction gets the best of both settings offering a more consistent definition of the transform.

While the correction is marginal for speech and music as most of the time the slopes are not big enough to produce a big shift, only when big pitch movements are utilized this becomes observable. This correction would become necessary in applications for which there are very high changes in fundamental frequency.

4.2 Chirp spread function in the $f - \alpha$ plane

Consider two chirps $C_1(t)$ and $C_2(t)$ with instantaneous frequencies $f_1(t)$ $t \in (t_{1s}, t_{1e})$, and $f_2(t)$ $t \in (t_{2s}, t_{2e})$. Let us consider that both chirps cross each other in the time–frequency plane. That is, there is some time $t_o \in (t_{1s}, t_{1e})$, $t_o \in (t_{2s}, t_{2e})$, for which $f_1(t_o) = f_2(t_o)$. There will be a sufficiently small interval around t_o for

which the instantaneous frequencies of C_1 and C_2 are similar. In this interval, the chirps can be approximated by stationary sinusoids as the frequency change would not be significant. That is: $C_1(t) \approx e^{j(2\pi f_1(t_o)t+\phi_1)}$ and $C_2(t) \approx e^{j(2\pi f_2(t_o)t+\phi_2)}$.

If we evaluate the inner product of both chirps $< C_1(t), C_2(t) >$, a non null value is obtained. The projection will take a value that is obtained essentially from the contribution of the projection in the surroundings of t_o. At times that are not close to t_0, the instantaneous frequencies of the chirps are different, so that the weight in the global scalar product result from those time sections will be negligible, as they are quasi-orthogonal.

Let us consider the chirp $C_1(t)$ as the signal of study, and $C_2(t)$ as a chirp used to evaluate the time–frequency representation at an arbitrary frequency f_2 and chirp-rate α, with an analysis window centered at time zero. There will be a set of values of f_2 and α for which the projection of C_1 on C_2 will not be close to zero. The projection will not be null when $f_1(0) = f_2(0)$ as they have the same central frequency regardless of α. But, the more different the chirp rates are, the shorter time the chirps capture energy from each other, and thus a smaller value will be obtained. It is important to note that this case of overlapping always produces a smaller value compared to the case where C_1 and C_2 have the same frequency and chirp rate, for which the maximum response is obtained.

Now we can observe the effect of the window length and shape in the $f - \alpha$ plane. The result of the projection will be affected by the value of the windows at the time of the chirps overlap. Shorter windows will attenuate the response when the overlapping occurs far from the center of the analysis window. Figure 4.3 shows a typical response of a sinusoid in the $f - \alpha$ plane for different windows.

4.2.1 Interference patterns in the $f - \alpha$ plane.

According to the analysis made in the previous section, the representation of a chirp in the $f - \alpha$ plane produces a point spread function with a 'butterfly' shape as can be observed in Figure 4.3. The representation does not concentrate all the energy in a reduced area in this plane, as the ideal would be to have only one spot in the whole plane. But the impact is not too strong for a single chirp as the representation still takes the maximum value for the actual chirp frequency and chirp rate, and it also does not exhibit other strong peaks for incorrect chip rates.

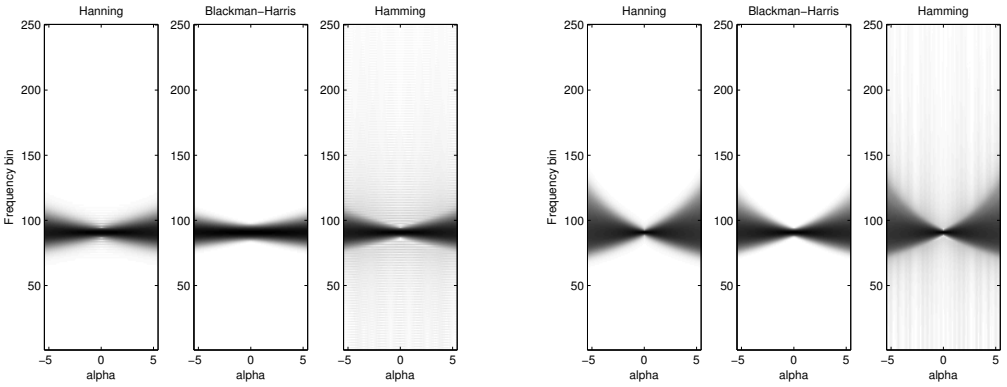


FIGURE 4.3: Example of the $F - \alpha$ plane for a sinusoid using linear warpings for three differing windows: Hanning, Blackman/Harris and Hamming. In the left the window length is 2048 samples and in the right the window length is 4096.

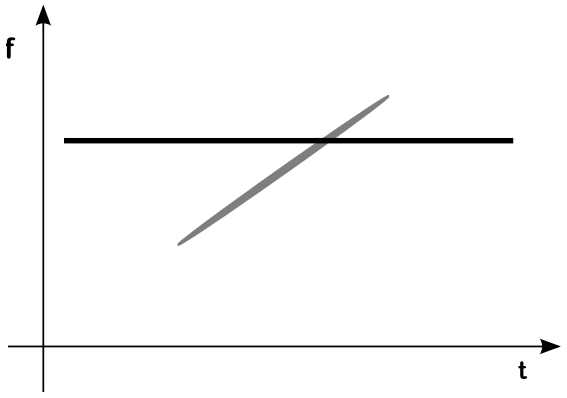


FIGURE 4.4: Schematic diagram of the crossing of a sinusoid and a chirp in the T-F plane.

The side effects of this spread of the energy becomes more relevant in the representation of two or more chirps. For the sake of simplicity let us consider that the signal to study consists in two sinusoids, with the expression:

$$x(t) = sin(2\pi f_1 t) + sin(2\pi f_2 t),$$

or only the positive side of the spectrum,

$$x_+(t) = e^{j(2\pi f_1 t)} + e^{j(2\pi f_2 t)}$$

Let us consider an analysis chirp $C_3(t)$ that sweeps a range of frequencies that include f_1 and f_2 within the analysis window support time. A schematic representation of the instantaneous frequencies of the signal and the analysis chirp is shown in Figure 4.5. There will be two overlapping regions as the one described

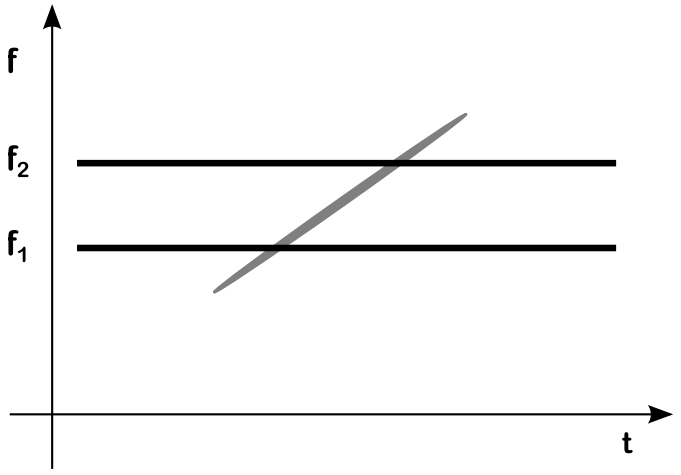


FIGURE 4.5: Schematic diagram of the crossing of the analysis chirp with two sinusoids in the T-F plane. When this crossings take place the interference patterns appear.

in the previous section. Each one will contribute to the value projection of $x(t)$ on $C_3(t)$. The main difference in this case is that depending on the phase of the signal and the approximate stationary sinusoid at the overlapping times, the resulting contribution can vary sensitively. If both phases differ in π, the result will be neglectful. If they have the same phase both terms will sum up and give a response that is twice as big as the case of a single chirp. A small change in the central frequency of C_3 or a slightly different α can make a considerable phase change at the overlapping points. This is especially true for higher frequencies, and stronger pitch change rates, as the absolute sweep frequency range of the chirps becomes broader. The result is the appearance of phantom peaks and valleys in the $f - \alpha$ plane, typically in the range of frequencies between f_1 and f_2, and a fairly different chirp rate from actual existing signal chirp rates. Figure 4.8 shows two sinusoids $x_1(t)$, and $x_2(t)$ that compose a signal $x(t) = x_1(t) + x_2(t)$. An analysis chirp $C_3(t)$ that sweeps the frequency range that includes f_1 and f_2 and using a Hanning window. The point-wise product of $x_1(t)$, and $x_2(t)$ with $C_3(t)$ is shown in the bottom graph. The picture shows that around the times the chirp crosses the signal sinusoids there is a contribution to the projection of $x(t)$ on $C_3(t)$, this is shown by low pass filtering the point-wise product $x(t).C_3(t)$ in order to show the local contribution to the final value.

The same result holds when the signal consists of two chirps that have the same pitch change rate and the analysis chirp has a significantly different α. This situation is quite common, as harmonic signals have many partials under this hypothesis. Particularly, every pair of consecutive partials of a harmonic source will show this side effect, becoming more evident in higher frequency ranges. This

can be observed in Figure 4.8, which shows the overlapping in time with the same phase and with opposite phases in time as well as the classical observed pattern that appears for that signal in the plane $f - \alpha$.

While the values of the artifacts are lower than those of the correct peaks that represent a study signal, many peaks appear at a wrong chirprate when a harmonic source is present.

4.3 Multi-resolution Fan Chirp Transform

4.3.1 Constant Q Transform

The constant Q transform provides a variable time–frequency resolution that offers a better representation in the overall spectrogram computed with the STFT. The Fan Chirp Transform also offers an improved representation when non stationary sinusoids are present. While the FChT overcomes some of the problems of using a larger window in higher frequencies, it still can benefit from the Constant Q Transform as the contour approximations are better in lower frequencies and rougher at higer frequencies. In this case, the quality factor that would be used is sensitively higher than a typical value in the classical STFT. We will base our analysis using the IIR-CQT [26], which gives a good compromise between computational cost and design flexibility, but similar results are obtained with other versions. See Chapter 2, for details on the IIR-CQT.

4.3.2 Constant Q and the Fan Chirp Transform

As mentioned before, the benefits of using a CQT transform instead of a fixed time window rely on the non stationarity of the signal, such as in the case of a human voice. The higher partials of a harmonic chirp are particularly non stationary, thus using a long window in the classical STFT may make their representation become blurred. The use of a CQT with an adequate value of Q makes it possible to achieve a better time–frequency resolution compromise for non stationary signals along all the spectrum.

Although the FChT using linear warpings is devised to analyze non stationary signals, the analysis may be further improved because of two main reasons. The first is that the range of fan chirp rates α used in the analysis is discretized. If

the signal chirp rate does not closely match any of the available α values, higher partials behave non-stationarely after the time warping. Using the CQT, with a high Q value, alleviates this problem. As a result, the number of analysis α values can be diminished, reducing the computational cost at no significant performance loss. The second reason is that considering a linear evolution of the instantaneous fundamental frequency could be a crude approximation for a signal frame with a non linear pitch evolution. In this case, the warping once again produces a slightly non stationary signal and the CQT is beneficial, especially for the higher partials (see figure 4.6). However, the addition of the CQT with a low quality factor, may produce some degradation in the analysis of a signal with a linear pitch evolution, so a relatively high Q value should be chosen in order to obtain a good performance for a wider set of signals.

Increasing the quality factor Q further may have the unavoidable drawback of increasing the interference patterns for incorrect frequencies and chirp rates, which may produce undesired artificial peaks in a polyphonic mixture. Figure 4.7 shows the $f-\alpha$ plane for some classical windows and using the constant Q transform to diminish the appearance of interference patterns in the overlapping areas of the partials point spread functions.

Figure 4.3 shows the response of four classical analysis windows in the $f-\alpha$ plane in the case of a sinusoid, and a linear chirp. Figure 4.7 shows the interfering effects that appear in a harmonic signal with 6 partials. The effect can be clearly seen in the overlapping areas of consecutive partials at big and small chirp rates.

4.3.3 Discretization of the linear warpings

The selection of the set of chirp rates that will be used in the linear FChT determines a lower bound of the approximation quality of the fundamental frequency contour.

If the analysis window time is too short, poorer concentration of the energy of the partials in peaks is achieved, but a smaller set of chirp rates α is enough to cover a wider range of chirp rates as the approximation is more local in time, and less local in the frequency and chirp rate dimensions.

In the case the analysis window time is too long, a big amount of energy concentrated in sharp peaks could potentially be achieved, but other effects limit this scenario and worse results may be obtained. Besides that the set of analysis chirps

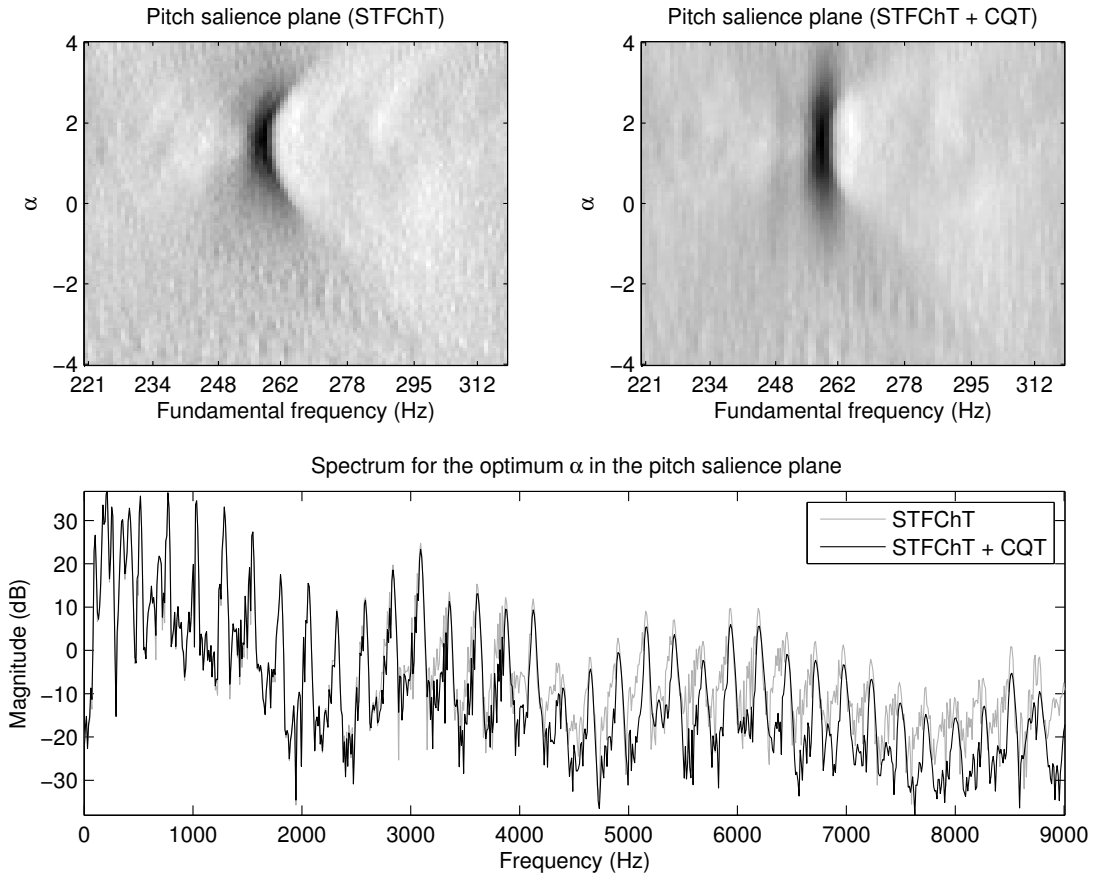


FIGURE 4.6: *Example of the FChT analysis including the CQT for a frame of the audio excerpt at $t = 2.68s$. The singing voice has a high pitch curvature at this instant. Adding the CQT enhances the representation of the higher partials. Note also that the peak region in the pitch salience plane is much concentrated in frequency and the range of α values with significant salience is widen which allows a more sparse α discretization.*

may have to grow considerably, which may only affect the computational time, the problem is that the linear approximation becomes weak for a long period of time. The analysis chirps may fail to be close to the actual signal's chirps, degrading the obtained resolution. Another issue is that the interference studied in Section 4.2.1 may become stronger, as a longer chirp may cross several partials with different frequencies and chirp rates. Also, the variation of the amplitude of each partial may influence the obtained resolution. If we assume that partials have a constant amplitude, the result is optimal, but in some cases, as real voice signals, the partials change their amplitude through time. This would appear as energy spread from the ideal desired peak, as a very long window would only capture mainly the mean amplitude of the partial. There is a tradeoff for which a medium length window the resolution of partials appearing as peaks is enhanced. An increase of the number of chirp rates is needed to represent accurately most fundamental

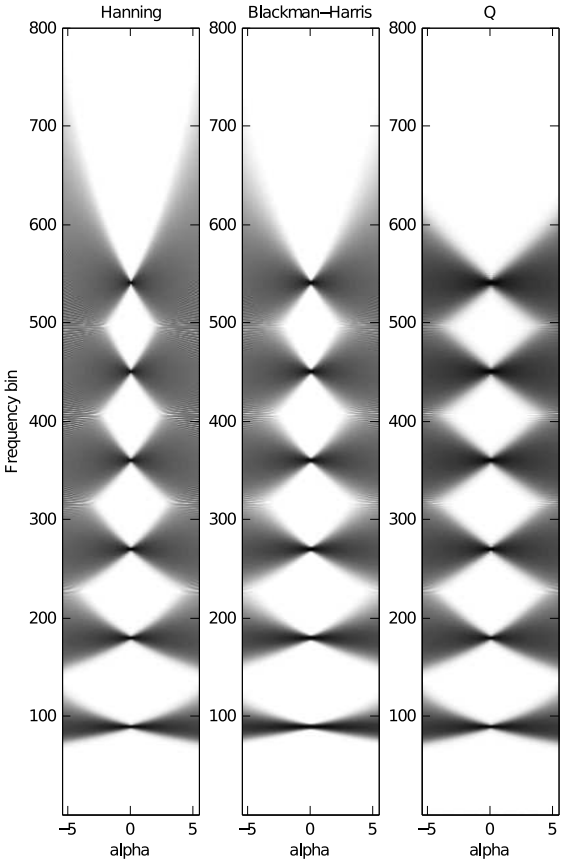


FIGURE 4.7: Harmonic signals. $f - \alpha$ plane represented with different analysis windows and Q transform. The overlapping areas where interference patterns can be seen are minimized with the unse of the Q transform.

frequency contours. In any case this length is much longer than the length that would be used in a STFT spectrogram.

A typical setup is using linear chirps and an analysis windows of around 100 ms, which corresponds to 4096 samples at a sampling frequency of 44100 Hz. In this case practical results show that about 11 chirp rates, in the range of values from -5 to 5 is enough to handle most pitch variations in a singing voice. If less than 11 slopes are used, the results start to degrade quickly while using more than 11 produces marginal improvements in the obtained resolution.

The combination of the constant Q-Transform with the FChT allows the use of wide windows for low frequencies, and mid-sized windows in higher frequencies. This combination offers a better compromise in the resolution as energy in low frequencies can be concentrated in very sharp peaks, and mid and high frequencies can be adjusted to obtain a good trade-off between the peak sharpness, and the ability to accurately approximate the F0-contour with a linear chirp close enough to the actual F0-contour change rate.

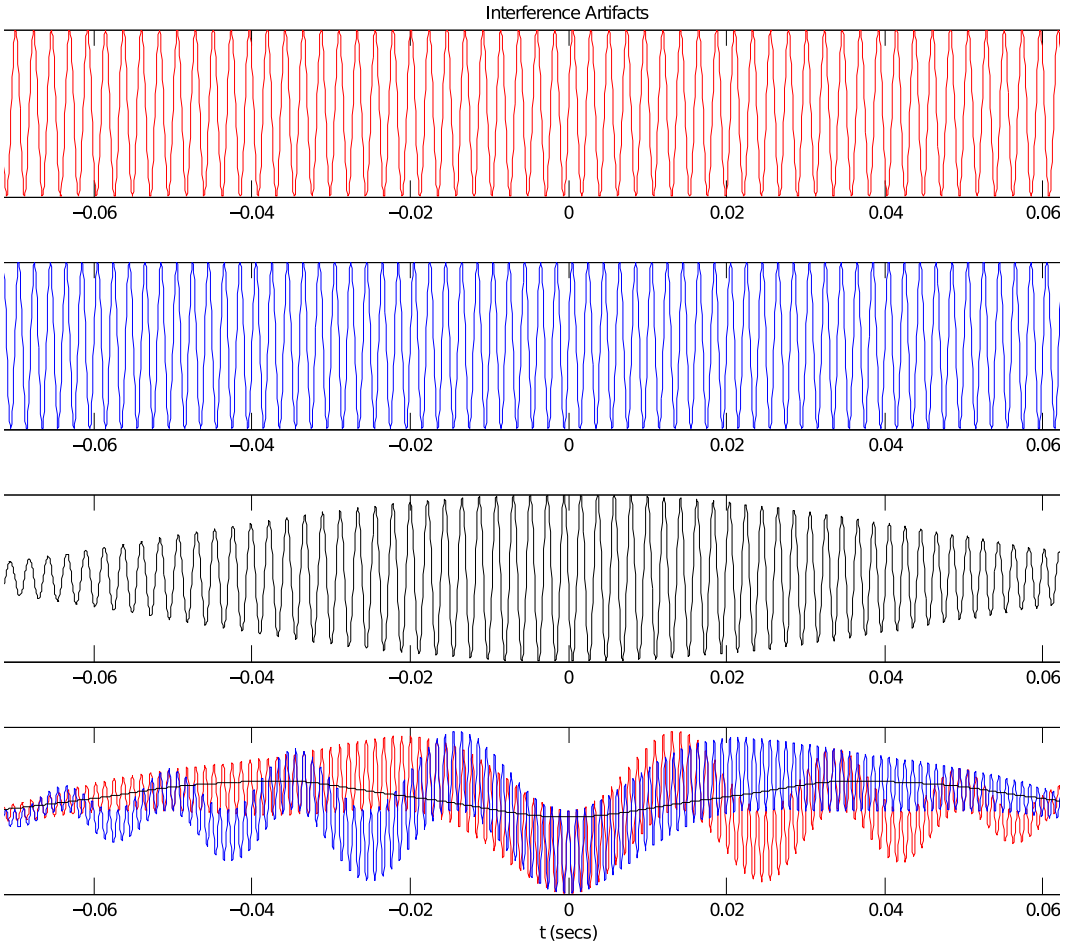


FIGURE 4.8: The two top graphs show two sinusoids $x_1(t)$, and $x_2(t)$ to be represented. The third graph is a chirp sweeping a frequency range that includes the frequencies of x_1 and x_2. In the bottom, the pointwise product $x_1(t) \times C_3(t)$ and $x_2(t) \times C_3(t)$ is shown together with the lowpass filtered $*x_1(t)x_2(t)) \times C_3(t)$ in order to show local contributions of the projection of the sinusoids on the chirp.

4.3.4 Non-linear warpings

The FChT uses a linear approximation of the variation of the fundamental frequency. In most cases a linear approximation is good enough to improve the representation of a signal when the right chirp rate is selected. But when the fundamental frequency has high curvature, the linear approximation may be rough. In this section we explore how the representation with Fan-Chirps can be extended to work with higher order approximations of the fundamental frequency contour.

4.3.4.1 Study of the distribution of the pitch contours

The linear FChT offers a good description of harmonic signals that have a relatively high pitch rate change. The transform assumes that a linear approximation is enough to accurately represent signals under this hypothesis. While this may be true most of the time when representing a singing voice, there are some F0 contours sections that cannot be accurately represented by a linear approximation. A good example of this are the vibratos, for which there are instants where the curvature of the F0 contours is high, and a linear approximation may become weak.

As the chirp rate functions can be arbitrarily defined, it is possible to construct a bigger set of warpings that correspond to a richer family of functions. One possibility is to consider not only linear, but also quadratic or higher order polynomial expressions to approximate the contours. A second approach would be to define the possible pitch variations from a database of real F0 contours. Given the analysis window time that is going to be used, we can take samples of the variation of the contours within its support time. Then, the contours' frequencies have to be normalized to make the representation of relative changes comparable. This can be done by simply dividing the contour by the frequency value at the center of the window. In this way a set of real relative F0-contours can be extracted and used to learn how to define the set of warpings based on statistical information. This can be done for example by taking the first two components of a PCA decomposition of pitch contours of an annotated database.

In the next two sections we will describe both approaches and compare the properties obtained from them.

4.3.4.2 Quadratic warpings

If second order functions are used to describe a fundamental frequency contour, a higher dimensional parameter space has to be considered, and now the values of slope and curvature have to be determined. The estimation of the ranges of values that those parameters can take can be based on the analysis of an annotated set of real data. The MIREX campaign [22] provides F0 contours as the ground truth for the training set of the MIREX Main Melody Detection Task. We will consider the quadratic approximation of the contours obtained by a minimum squares polynomial fitting of the form $1 + \alpha t + \beta t^2$ for a set of segments of the F0-contours with the defined window length. A histogram of the pair of coefficients α

and β, that represent the slope and curvature, is constructed and shown in Figure 4.9 . Some practical considerations have to be taken into account to correctly calculate the coefficients. The first is that the window time length has to be defined accordingly to the time window that will be used in the analysis of the FChT. As aforementioned, the contour has to be scaled to have the same frequency at the center time of the chosen window in order to represent relative frequency changes. As the FChT models properly relative variations of pitch, the only way to make a valid study is to normalize them to be comparable. As an arbitrary choice, the frequency value chosen to be set at the central window time is 1. In this way the contour segment values are exactly the relative frequency. The expression that relates the actual F_0 contour is

$$F_0(t) \approx F_0(t_c)(1 + \alpha(t - t_c) + \beta(t - t_c)^2), \text{ with } t \in (t_c - t_w/2, t_c + t_w/2)$$

where t_c is the time at the center of the window where the approximation holds, and t_w is the analysis window length.

Figure 4.9 shows the two-dimensional histogram of α and β for the complete set of MIREX04 and MIREX05 datasets. As it can be seen, there is a considerable amount of the contours with a small or null α and β, this corresponds to no frequency change which is the most common situation. The histogram can be used as a tool to determine the region of pairs of values of α and β that is reasonable to consider to cover most of the frequent values. Linear chirps are considered correspond to points in the histogram with $\beta = 0$, so they are samples along the horizontal axis. Some examples of different samplig of this parameter space and representation examples can be checked in [30].

4.3.4.3 PCA-based warpings

Another approach is to approximate the variations of the frequency contours in terms of a linear combination of two or more functions learned from the data. These functions should concentrate as much energy as possible of the observed variations. A natural way of finding these variations is to choose the first components of a PCA decomposition. As in the quadratic approximation setting, it is also necessary to scale the contours in order to have the same central frequency, also chosen as 1.

The first three components of the PCA decomposition calculated from the MIREX04 and MIREX05 databases are shown in Figure 4.10. In order to make the analysis,

the relative frequency values are calculated for a window time of $100ms$, and the center value is substracted to capture only the variations and not the mean value. The first component is a function that resembles a linear change, and the second captures most of the quadratic variation information presenting with no slope and curvature at the center of the window. The main difference with the quadratic approximation appears in the ends of the analysis time window, where the variation is softened, not following the trend of the same linear or quadratic change but showing a milder deviation from the center frequency. This can be interpreted as that the variations in the fundamental frequency of a melody or a singing voice have a local nature, and they are typically not sustained for a long time. The decomposition in PCA components would permit a slightly better representation of the set chirp rates in a minimum squares sense as the terms are learned from the data.

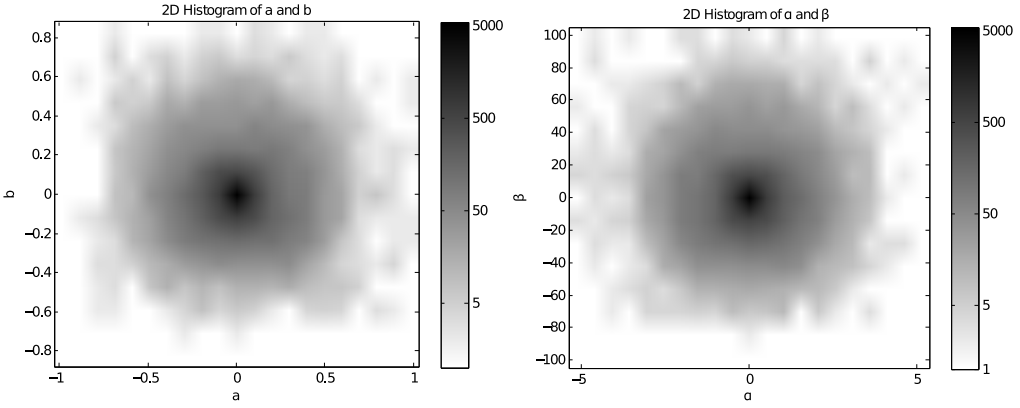


FIGURE 4.9: Left: Histogram of the coefficients (a,b) of the PCA first two components for MIREX04 and MIREX05 databases. Right: Histogram of the polynomial coefficients α and β for the same database.

For each time t_c, there will be a pair of coefficients that produce an approximation of the F_0 contour. The expression that relates the actual F_0 contour is

$$F_0(t) \approx F_0(t_c) + aC_1(t - t_c) + bC_2(t - t_c), \text{ with } t \in (t_c - t_w/2, t_c + t_w/2)$$

where t_c is the time at the center on the window where the approximation holds, t_w is the window time length (0.1s in our study), and a and b are the projections of the contour in the components $C_1(t)$ and $C_2(t)$.

It is also possible to generate a histogram with the frequency of the values of the composition of F0 contours. This can be done by projecting each contour in a window time on the selected components, and computing a histogram of the pair of representation coefficients. Figure 4.9 shows the histogram of the pair of values

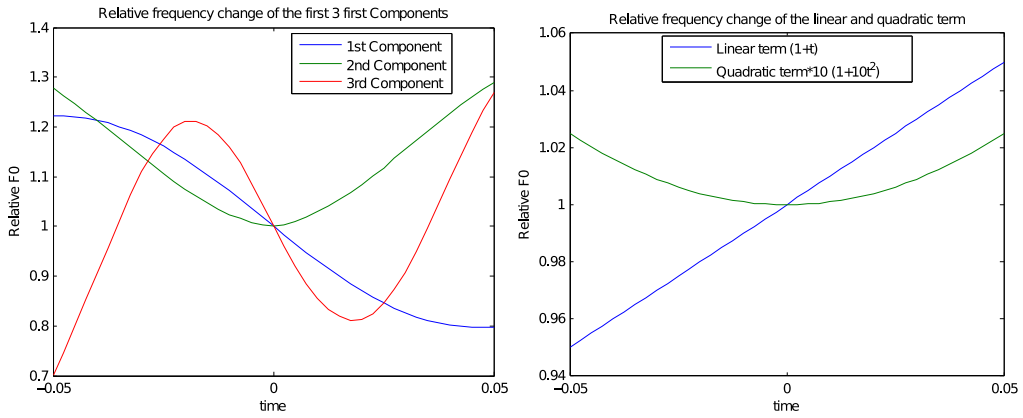


FIGURE 4.10: Left: F_0 relative variaton for the three first PCA components obtained from the MIREX04 and MIREX05 databases. Right: for comparison purposes, the linear and quadratic terms frequency variation are ploted, the quadratic term is scaled by a factor of 10 to make the comparison easy.

a and b. The distribution in the histogram is similar to the one obtained in the quadratic case, but the values are slightly better concentrated in a smaller range, which will be reflected in less samples in this space to represent a similar set of possible frequency contours with the same mean squared error.

4.3.5 FChT with non-linear warpings

Different approaches can be followed to define the sampling of the space of parameters that permit a good coverage of the different time warpings found in real data. The linear FChT can be considered as taking a set of samples along the horizontal axis. While it is a good initial approximation, a more comprehensive sampling can be defined considering also points that involve not only the slope but also the curvature of the contour. In [30] we defined a set of samples in the parameter space to cover the area of the histogram with the greater number of occurences. As most of the time the contours have low curvature, there is no improvement in the representation for them, or there could be a slight degradation in those cases as the new warping could correspond to new intereference patters as the ones described in Section 4.2.1. The improvement in the representation for this sampling is restricted to times in which the frequency contour has medium or high curvature. A good example of a singing voice under these hypothesis is in Opera as many vibratos are present. Figure 4.11 shows an example of a real Opera signal from the MIREX04 database; in the maxima and minima regions of the vibrato a slight improvement can be observed when using quadratic warpings compared to using only linear warpings. Similar results are obtained with the

PCA based nonlinear warpings as the main differences in the frequency changes are in the ends of the analysis time, where typically the analysis window takes small values (i.e. Hanning).

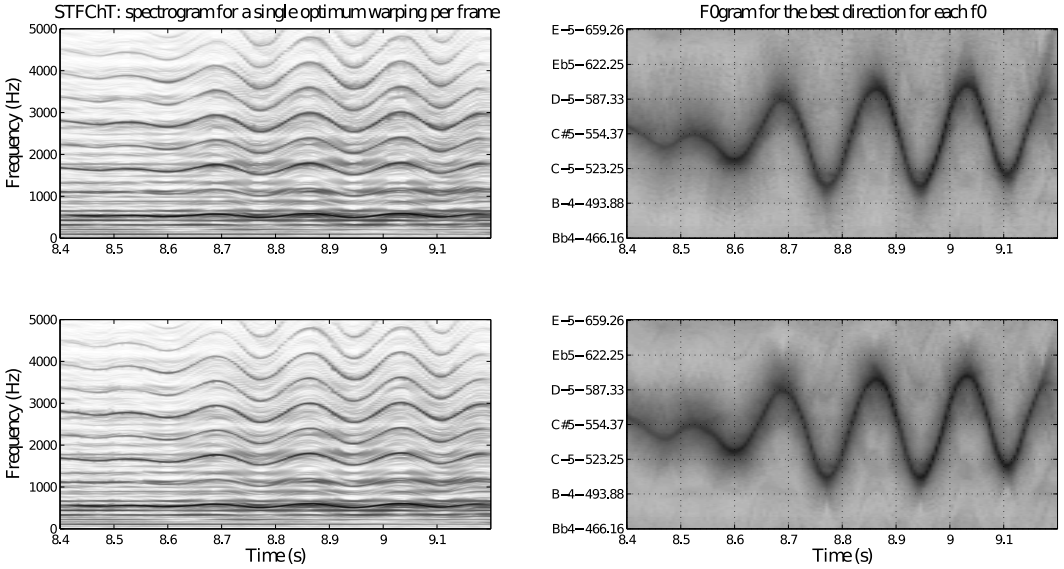


FIGURE 4.11: Comparison of the FChT computed with linear warpings (top), and a set of quadratic warpings (bottom). A slight improvement can be seen at the times where the contours have a high curvature.

In order to comprehend all the probable pitch evolutions, a set of samples in the parameter space has to be selected. The design criteria would define a grid of points in the $\alpha - \beta$, and $a - b$ plane that covers most of the dense histogram areas. As the number of total samples should be kept as small as possible to make the computational cost reasonable, there is a trade-off between this computational cost and the density of the sampling.

4.3.6 Conclusions and Future Work

Different extensions and limitations of the FChT were analyzed. The combination of the FChT with a constant-Q transform with a big quality factor shows some benefits as the linear approximation becomes more appropriate for high harmonics of a singing voice.

All the extensions provide slight improvements of the linear FChT representation. We believe that the FChT with the correct chirp rate is already close to the limit of the uncertainty principle as the representation tool is strongly adapted to the signal. This gives little room for big enhancements in the representation when applying both the nonlinear warpings and using a constant quality factor. The

fact that they can be combined together in a modular way makes the approach simple and easy to understand.

As the analysis with the FChT and the described extensions are calculated in a local way, there are some interfering terms that appear as the signal can be locally interpreted as having a completely different fundamental frequency and chirp rate as the actual one. This ambiguity could be eliminated if information from contiguous frames would be used. While this is possible, it can be seen as an advantage to define the transform as a low level local processing and let higher level applications decide the coherence of each possible solution given its context. Besides this, we think that it is important to describe the nature of the spurious components that appear in the local representation.

Chapter 5

Applications of Time–Frequency Representation Tools

In this Chapter we describe the different MIR applications that were approached. The next section describes a pitch content visualization tool. After that we present some results of an algorithm that tracks the main melody based on the F0gram, that was submitted to MIREX. We also discuss an algorithm that exploits the chirp rate information to perform the tracking based on a clustering technique.

5.1 Pitch Content Visualization Tools For Music Performance Analysis

In this Section we describe the use of the FChT as a tool to assist the musical analysis of a piece. The tools that aided the analysis were released as an open source and freely available plug-in for a general purpose audio visualization tool, as well as a MatLab library.

The analysis is based on the publication [21], in which the software was made available. The software was developed in collaboration with Ernesto López, Martín Rocamora, Haldo Spontón and Ignacio Irigaray and the musicological analysis of music was carried out by Luis Jure. The section reproduces some the article passages, as well as includes some modifications or additions in order to contribute to the structure of this document.

5.1.1 Introduction

Most techniques for musical analysis do not work directly on the acoustic signal, but on a symbolic representation of it [31]. The development of new musical analysis techniques based on the spectrogram began to be explored systematically with the work by Robert Cogan [32]. Using time–frequency representations of the audio signal; Cogan proposes an analytical method applicable to both structural and local aspects of a musical piece that exemplifies analyzing music from very varied corpus. Recently, techniques based on graphical representations have been applied extensively to the analysis of electroacoustic music [33]. These tools are also being applied to notated music or music from traditions not based on scores, to discuss aspects of music not represented in symbolic notation by the analysis of recordings. This may include components that depend both on the performance [34] (such as temporal and tuning micro-deviations), or on the precise determination of the tuning system of a certain music [35].

Different software tools for computer-aided analysis, visualization and annotation of recorded music have been developed, for instance Sonic Visualiser.[1] They typically include traditional time–frequency representations and digital signal processing tools intended for music information retrieval, such as onsets or pitch detection. Some mid-level representations are also available, i.e., signal transformations that tend to emphasize higher semantics than the energy in the time–frequency plane [36]. Those mid-level representations are usually devised to facilitate the subsequent feature extraction and processing of an automatic algorithm. However, as suggested in [37], they can also be used by humans to study performance nuances such as pitch modulations or expressive timing.

In this Section, two software tools to calculate the F0gram, based on the Fan Chirp Transform (FChT) are presented. These tools seek two main goals: firstly, the precise time–frequency location of the components of a complex sound, using recent analysis techniques that overcome the limitations of the classical tools; secondly, the automatic grouping of all the components that are part of the spectrum of a single harmonic source, highlighting the fundamental frequency, f_0. This makes possible to obtain an accurate graphical representation of the temporal evolution of the melodic content of a music recording, that allows for the detailed study of performance aspects related to pitch intonation and timing (e.g. tuning system, vibrato, glissando, pitch slides).

[1] http://www.sonicvisualiser.org/

5.1.2 Time–Frequency analysis

Music audio signals often exhibit ample frequency modulation, such as the typical rapid pitch fluctuations of the singing voice. Precisely representing such modulations is a challenging problem in signal processing. It is reasonable to look for a signal analysis technique that concentrates the energy of each component in the time–frequency plane as much as possible. In this way, the representation of the temporal evolution of the spectrum is improved and the interference between sound sources is minimized, simplifying the task of higher level algorithms for estimation, detection and classification.

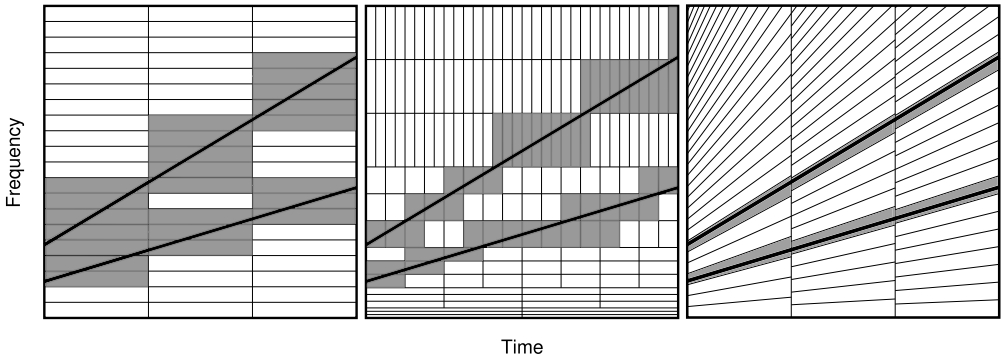


FIGURE 5.1: Time–Frequency tiling sketch for the STFT, the Short Time CQT and the Short Time FChT and the resulting resolution for a two-component harmonic linear chirp.

As we mentioned before, the standard method for time–frequency analysis is the Short Time Fourier Transform (STFT) which provides constant resolution in the time–frequency plane. A typical alternative for multi-resolution analysis is the Constant Q Transform (CQT). Both representations produce a Cartesian tiling of the time–frequency plane, as depicted in Figure 5.1 which does not give the best results for non-stationary signals, for instance a frequency modulated sinusoid or chirp. The virtue of the FChT is that it offers optimal resolution simultaneously for all the partials of a harmonic linear chirp, i.e. harmonically related chirps of linear frequency modulation. This is well suited for music analysis since many sounds have a harmonic structure and their frequency modulation can be approximated as linear within short time intervals.

For polyphonic music analysis there is no single optimal chirp rate α value, so several FChT instances with different α values are computed, as described in Section 3.3.4. This yields a multidimensional representation made up of various time–frequency planes. The selection of the α values that produce the better representation of each sound present is performed by means of pitch salience. A

comparison of the STFT and the STFChT applied to a polyphonic music audio clip is provided in Figure 5.2.

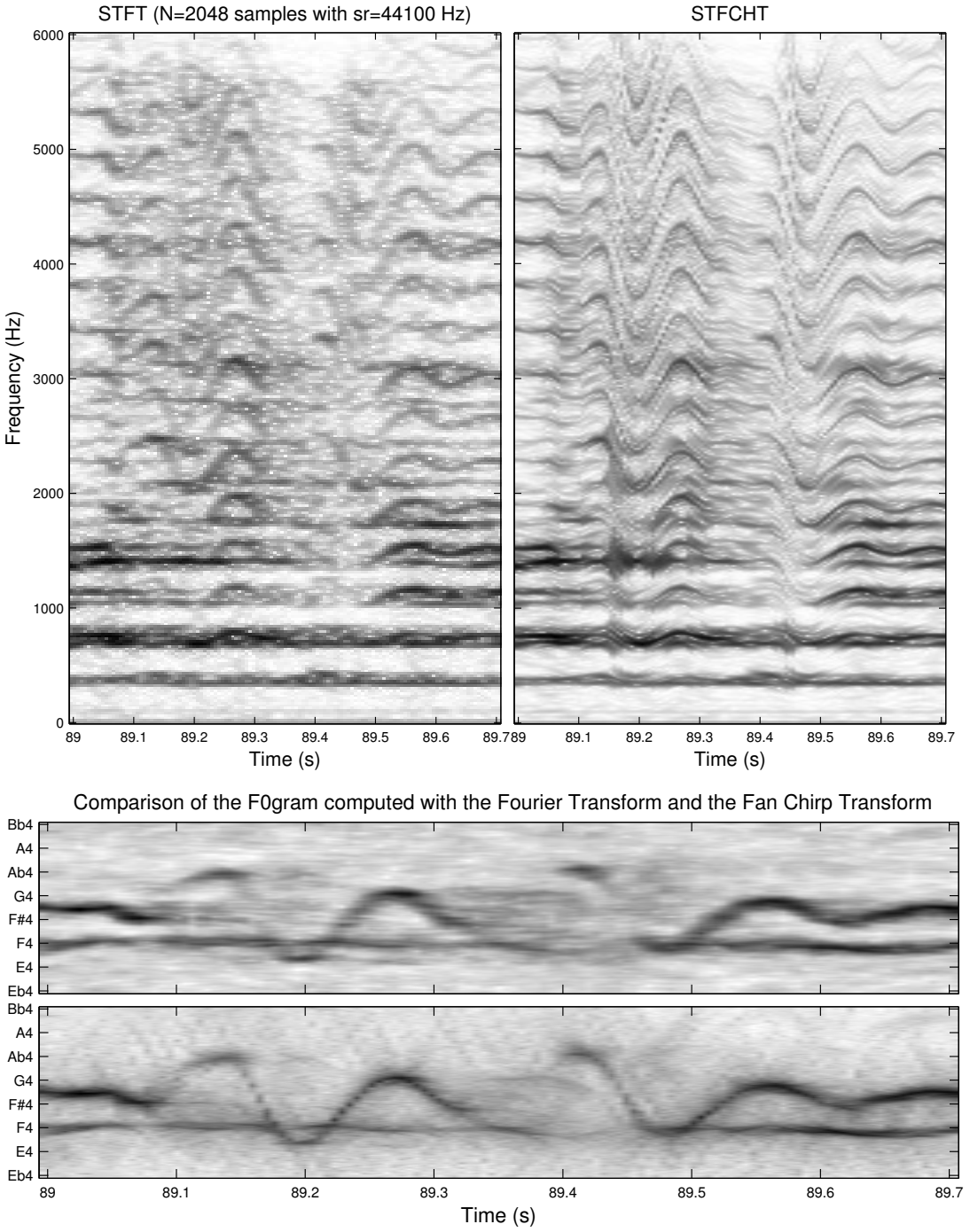


FIGURE 5.2: Above: Comparison of the STFT and STFChT for an excerpt from the example of section 5.1.4.1. The chirp rate of the most prominent sound source is selected for each frame. Note the improved representation obtained for this source while the rest is blurred. Below: F0grams obtained from the DFT and FChT. Rapid pitch fluctuations are better represented in the latter.

5.1.3 Pitch salience representation

A representation intended for visualizing the pitch content of polyphonic music signals should provide an indication of prominence or salience for all possible pitch values within the range of interest.

Given the FChT of a frame $X(f, \alpha)$, salience of fundamental frequency f_0 is obtained by summing the log-spectrum at the positions of the corresponding harmonics, as described in Section 3.3.1 by,

$$\rho(f_0, \alpha) = \frac{1}{n_H} \sum_{i=1}^{n_H} \log |X(if_0, \alpha)|, \tag{5.1}$$

where n_H is the number of harmonics located up to a certain maximum analysis frequency. This is computed for each signal frame in a certain range of f_0 values.

Some postprocessing steps are carried out in order to attenuate spurious peaks at multiples and submultiples of the true pitches, and to balance different fundamental frequency regions. Finally, for each f0 in the grid, the highest salience value is selected among the different available α values. In this way, a representation that shows the evolution of the pitch of the harmonic sounds in the audio signal is obtained, namely an F0gram. Refer to Section Section 3.3.1 for more details. Examples of the resulting representation are depicted in Figure 5.3 for two short audio clips. The F0gram produces a fairly precise pitch evolution representation, contrast balanced and without spurious noticeable peaks when no harmonic sound is present. Note that simultaneous sources can be correctly represented, even in the case that they coincide in time and frequency if their pitch change rate is different, see panel A of Figure 5.3. Figure 5.2 shows a comparison of the F0gram obtained from the DFT and the FChT. The improvement in time–frequency localization provides a more accurate representation of pitch.

5.1.4 The F0gram as a tool for musicological analysis

Two case studies are presented to exemplify how the F0gram can be used as a powerful tool to do some musicological analysis that would be hard to do with more classical tools as a simple spectrogram. The selected case studies include vocal parts exhibiting complex pitch evolution, very difficult or downright impossible to notate with precision using Western common music notation: a field recording of a folkloric female vocal trio from west-central Bulgaria, and a commercial recording

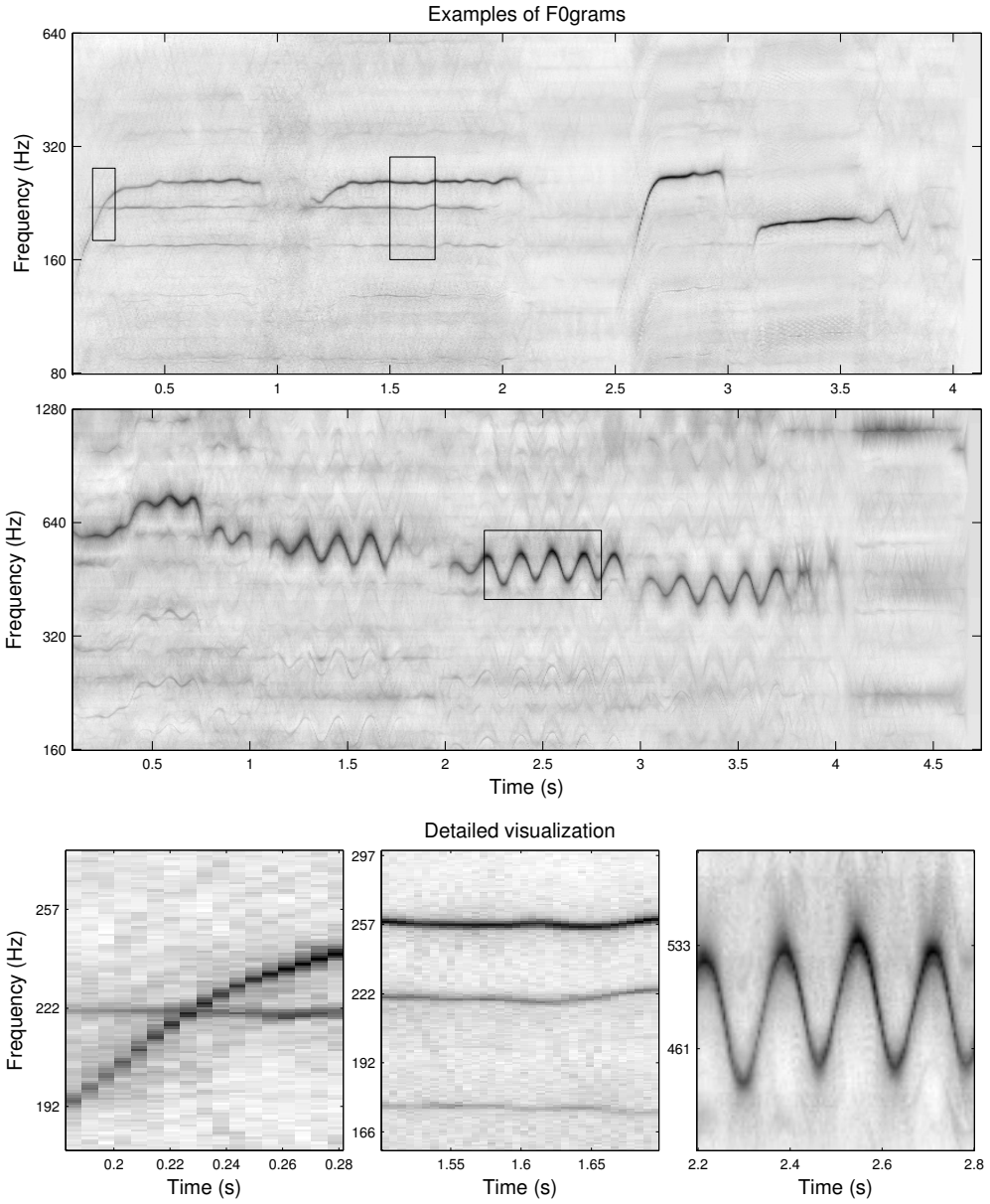


FIGURE 5.3: Above: F0gram examples for audio excerpts of *pop1.wav* and *opera_fem4.wav* from the MIREX melody extraction test set. Below: Detailed visualization. A) Crossing pitch contours are well resolved, B) and C) simultaneous sources and rapid pitch fluctuations are precisely represented.

by noted blues singer and guitarist Muddy Waters. In the next two sections we summarize the analysis for each case, indicating the properties of the F0gram as a tool that made possible an analysis that would have been very difficult to do without it. For more details on the musicological analysis performed by Luis Jure, refer to [21].

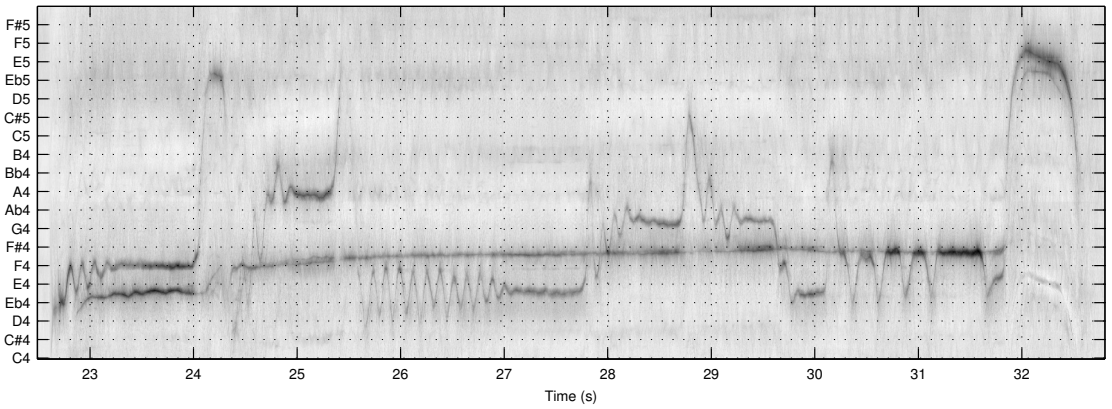


FIGURE 5.4: One phrase of the Harvest Song, showing characteristic traits of Shope diaphonic singing: melodic ornamentation in the first part (including *tresene* and a final *izvikvane*), and a pedal on the "tonic" in the second part, with a slow glissando.

5.1.4.1 Case of study: Folkloric Diaphonic chant of the Shope country

The F0gram was used to analyse a Bulgarian polyphonic folkloric song. As a general rule, these polyphonic songs are performed by female singers, and the sound of the voices itself is usually enough to impress listeners not familiar with this idiom.

In a typical setting of this song, the melody part is sung by one singer, and the second part by two or sometimes more. The upper part has several classified melodic gestures. The second part is more static, and can be described as a "drone" or pedal. It usually stays on the tonic of the mode, with occasional deviations to the sub-tonic when the melody descends to the tonic. Both parts join, however, to perform a gesture called *izvikvane* together. Apart from some fast swoops, the melody part moves within a very limited range. This results in a preponderance of narrow intervals between the voices [38, 39].

For our case study, a commercially available field recording of a folkloric group from the Shope region was used [40]. The recording is identified, without further information, as a "Harvest Song" performed by a female vocal trio from the village of Zheleznitsa. The recording date can be placed around 1980. The song consists of 9 short phrases of similar duration (ca. 10~12 s), structured as three variations of a group of three distinct phrases.

Figure 5.4 shows an F0gram of the third of these phrases, exhibiting all typical characteristics of this chant. The cadential *izvikvane* covers a narrow octave before descending back to the tonic area, and sounds like a unison of the three

voices, in accordance with the prevailing description of *izvikvane*. The F0gram allows us to appreciate, however, that there is actually a slight separation of the voices, very difficult to perceive by listening alone.

An analysis of the simultaneities confirms that narrow intervals prevail, and a variety of intervals can be found between the unison and the major third. The F0gram obtained from the FChT permits a precise measurement of these type of intervals, as can be appreciated in Figure 5.2.

Of special interest was the location of the sub-tonic, and the interval most frequently found lies half-way between one and two semitones below the tonic (sec. 23–24). This same kind of "second" can often be found above the tonic, in the upper part (seconds 28–29). The speed and range of the *tresene* can also be assessed with good precision. Typical rates are around 8~9 Hz (sec. 26–27), but slower rates can also be found (sec. 30–31). The width is variable, extending through intervals of up to three semitones (sec. 26).

The analysis of the F0gram permits a better measurement of the characteristics described. Observing the second part with more detail, the "pedal" does not remain on a fixed note but performs a slow upward glissando, covering roughly the equivalent of a semitone in approximately 7 seconds.

An analysis of the contours that can be found in the F0gram permits to observe that unlike a typical drone or pedal point, essentially static, it has a dynamic character, and this kind of slowly ascending movement imposes on the polyphony a very particular tension and expressiveness. Additionally, in the F0gram it can be clearly noticed that the "tonic" varies between phrases. More details on the musicological analysis can be found in [21].

This phenomenon is not mentioned in the consulted bibliography and is not represented in the available transcriptions, although it was found in various degrees in several recordings analyzed, suggesting they are part of the genre and should be considered an essential component of the powerful expressiveness of this particular form of folkloric polyphonic singing. To some extent we believe that this type of analysis becomes possible with tools as the presented here, and would be significantly more difficult with classical spectrogram representations.

5.1.5 Case of Study: Muddy Waters - Long Distance Call

The second case of study comprehends the Blues genre of popular music deeply rooted in the African-American folksong tradition of the rural South of the United States. The reason is because some of the gestures in this music involve pitch contours that exhibit special variatons that are characteristic of the genre. The most representative example of these are the so-called *"blue notes"*, which precise definition has been elusive and even somewhat controversial.

Rather than fixed tones in a discrete scale system, blue notes would be flexible areas in the pitch space. For the analysis of the behaviour of these pitch complexes in actual performance, we chose a recording by Muddy Waters, one of the most important blues musicians.

The Muddy Waters song analysis is based on is a recoring made on January 23, 1951 when he recorded his own composition "Long distance call" for Chess Records. He sings and plays electric guitar, and is accompanied by Marion "Little Walter" Jacobs on harmonica and Willie Dixon on double bass. The song is a standard 12-bar, three-line stanza blues, where the second line in each verse repeats the first, and the third is a rhyming conclusion. After a 4-bar introduction, the first stanza extends from measure 5 to 16. Figure 5.5 shows the F0gram and the transcription of the six measures where Muddy Waters sings the lyrics: mm. 5-6, 9-10 and 13-14. Each of these 2-bar vocal phrases is followed by a 2-bar instrumental response, omitted in the figure.

As in the previous case of study, more details on the musicological analysis can be found in [21]. The application of the proposed analysis tools permits a clear visualization of two salient traits of this passage: a melody consisting mostly of time varying pitches with relatively few moments of stability, and the establishment of continuous tonal regions non reducible to single pitches in a discrete scale.

5.1.6 Discussion

By means of the analysis of two music recordings, the usefulness of the introduced techniques for computer aided musicology was illustrated, in particular for discussing expressive performance nuances related to pitch intonation. The result of the analysis by itself reveals important aspects of the music at hand, difficult to asses otherwise. The computational techniques implemented are oriented towards the precise representation of pitch fluctuations.

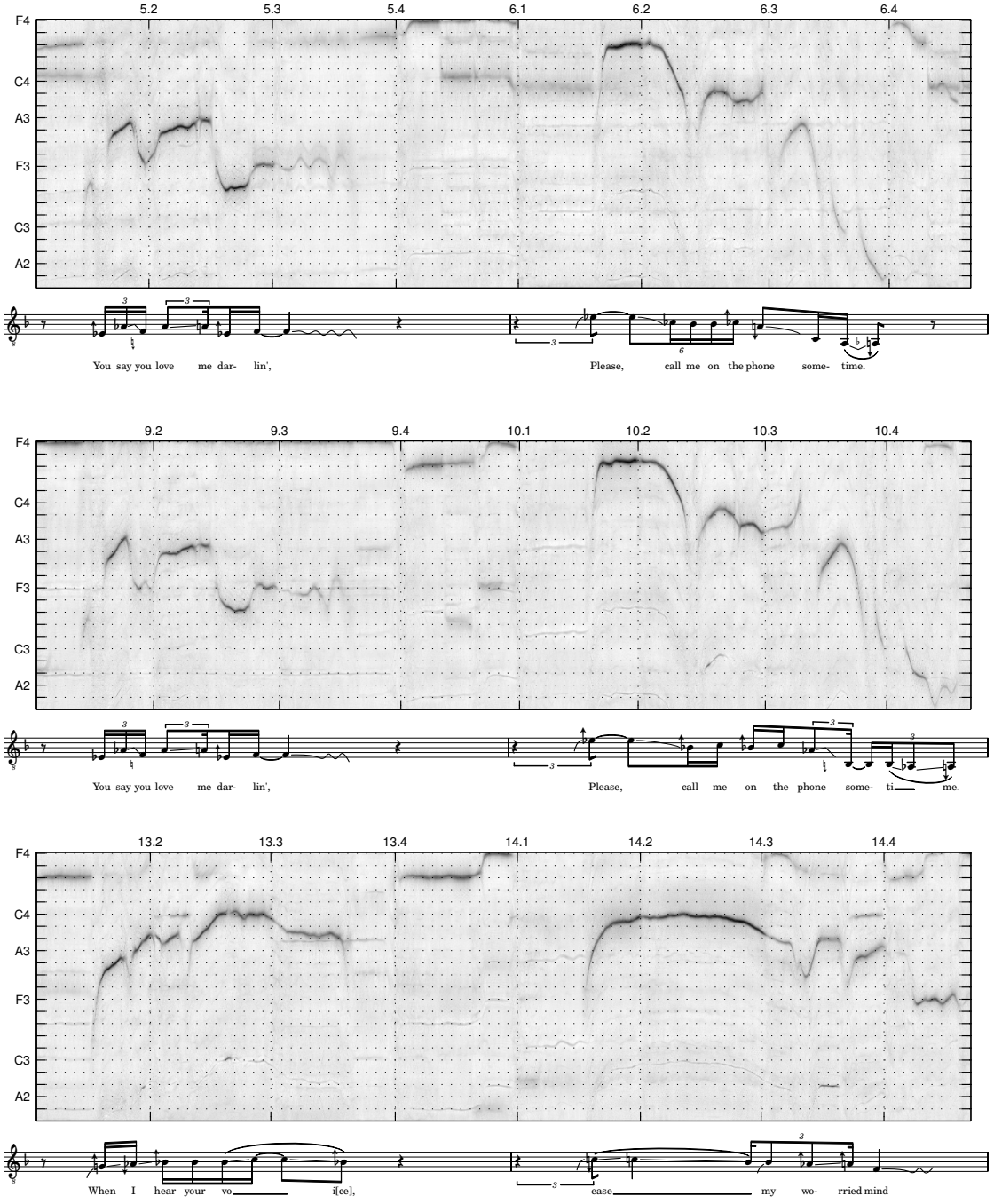


FIGURE 5.5: Muddy Waters, "Long distance call" (1951): F0gram showing continuous pitch contours of the voice, and approximate musical transcription informed by the analysis of the F0gram.

Two graphical software tools were developed to allow the application of the described methods by the research community.[2] One of them is a Vamp plugin[3]

[2]Available, as well as the audio clips, from http://iie.fing.edu.uy/investigacion/grupos/gpa/ismir2012

[3]http://www.vamp-plugins.org/

for Sonic Visualiser that computes the pitch contours representation. Within this application several other features are available that can assist the analysis. A Matlab® GUI is also released that includes additional functionalities and information, better suited for signal-processing researchers.

5.2 Pitch Tracking Applications

5.2.1 Introduction

Pitch Tracking in polyphonic music is a challenging problem in the Music Information Retrieval field. The F0gram based on the FChT representation produces a set of candidates of the possible pitches present at a given time that can be exploited by a tracking algorithm to obtain the F0 contours. Some works have already explored this idea in speech [41] with good results.

Two different algorithms were explored based on the FChT representation to track the melody frequency contour in music. The first one is an Ad-Hoc tracking algorithm based in the F0gram to find the Main melody in the MIREX [22] Main Melody Extraction Task. The second one is based in a clustering technique and tries to exploit the chirp rate information that can be obtained from the FChT representation. This Section is based on the publication [42], in collaboration with Martín Rocamora. The description reproduces some of the article passages, as well as includes some modifications or additions in order to contribute to the structure of this document.

5.2.2 Main Melody Extraction

A frame based melody detection evaluation was conducted to asses the usefulness of the proposed FChT-based method for music analysis. To do this, two different labeled databases were considered, namely the 2004-2005 MIREX [22] melody extraction test set (only vocal files) and the RWC Popular Music database [29]. The former comprises 21 music excerpts while the latter contains 100 complete songs, for a total duration of 8 minutes and 6 hours respectively. The RWC is a more difficult dataset due to higher polyphony (including prominent percussion) and dynamic compression, whereas the MIREX although much smaller is publicly available and more diverse (e.g. includes opera). For each frame the most prominent F0gram peaks were selected and their corresponding fundamental frequencies were considered as main melody pitch candidates. Only those frames for which the melody was present according to the labels were taken into account to compute the evaluation measure according to,

$$\text{score}(f_0) = \min\{1, \max\{0, (\text{tol}_{\max} - \Delta f_0)/(\text{tol}_{\max} - \text{tol}_{\min})\}\}$$

where $\Delta f_0 = 100|f_0 - f_0^{gt}|/f_0^{gt}$ is the relative error between a candidate and the ground truth, and the tolerances tol_{max} and tol_{min} correspond to 3% and 1% respectively. This represents a strict soft thresholding of the estimation performance[4].

Considering that the pitch of a main melody is not equiprobable in the f_0 selected range, it is reasonable to include this a priori information in the selection of candidates. To do this, salience is weighted by a gaussian centered at MIDI note 60 (C4) and with a standard deviation of an octave and a half. This values were selected considering the main melody pitch distribution of the databases (see figure 5.6), but setting the model parameters to favour generalization (in particular tripling the standard deviation).

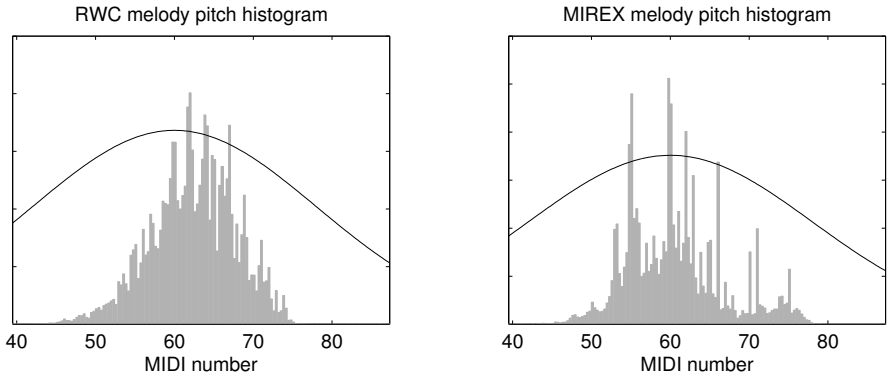


FIGURE 5.6: Pitch preference function (mean = 60, stdev = 18) and melody pitch histogram for RWC Popular and MIREX data.

In figure 5.7 an example of the melody detection result is depicted. Note that the first candidate correctly follows the main melody when it is the most prominent. The fan chirp rate estimated for the first candidate matches the actual value extracted from labels, as shown in figure 5.8. This information can be further exploited, for example when performing the temporal tracking.

Table 5.1 shows the scores obtained for the different methods when applied to each database. The optimal parameter values were grid searched for every method. It turned out that for the FChT-based methods similar results were obtained for parameters around the values specified in section 3.1.2 and different number of α values. Results reported correspond to 15 fan chirp rates. In this case, running

[4]Note that this performance measure is stricter and more discriminative than the binary 3% threshold used in MIREX. This was devised in order to better distinguish the performance of the different methods evaluated and considering that the ground truth labels are not perfectly accurate.

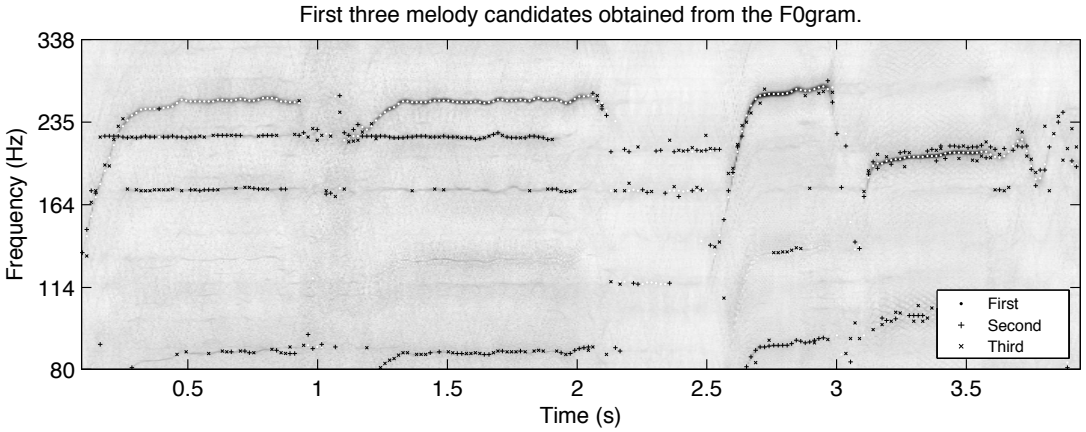


FIGURE 5.7: Example of melody detection. The three first candidates and a ±3% band centered at the label are displayed. First candidate tends to correctly match the melody while the remaining ones usually chose other secondary voice. Note that not attenuated submultiples mislead the detection of these secondary sources.

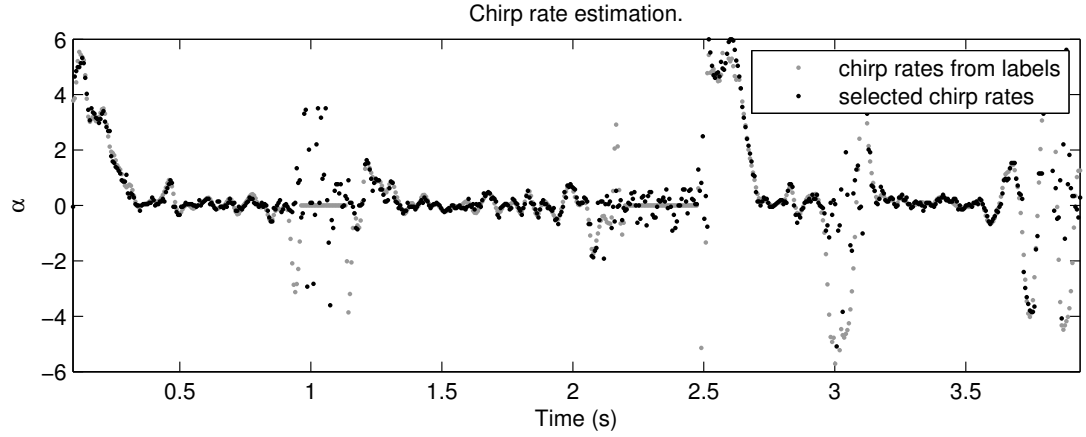


FIGURE 5.8: Estimated fan chirp rate for the first candidate and actual value extracted from labels.

time in a 2.5 GHz core i5 processor reaches real time for a Matlab and C code implementation of the FChT [5].

The results indicate that the proposed improvements to the pitch salience computation (submultiple attenuation, normalization and a priori model) contribute to a significant performance increase (compare the two STFT results). Further performance improvement is achieved by the use of the FChT. Although the combination of the FChT and the IIR-CQT leads to a better representation of some frames, its impact into the melody detection results is marginal. This is related to the fact that higher partials are less influential in the pitch salience value given

[5]The implementation is available at
http://iie.fing.edu.uy/~pcancela/fcht.

their lower amplitude. Additionally, the number of frames with highly non-linear pitch evolution represent a small amount of the total.

TABLE 5.1: *Melody detection scores (%) from 1st to 5th candidate. Except for the first method where only multiple attenuation is applied, all the others include the proposed improvements: submultiple attenuation, $\rho_2(f_0)$ normalization and f_0 preference function.*

MIREX	**STFT plain**	**STFT**	**FChT**	**FChT +CQT**
1	71.32	75.72	81.92	**82.09**
1-2	78.90	82.82	87.32	87.41
1-3	82.63	85.95	89.67	89.82
1-4	85.14	87.71	91.09	91.24
1-5	86.80	88.92	92.11	92.19

RWC	**STFT plain**	**STFT**	**FChT**	**FChT +CQT**
1	48.60	63.19	68.21	**68.52**
1-2	59.50	72.96	76.85	77.50
1-3	65.62	77.53	80.81	81.55
1-4	69.86	80.33	83.27	84.07
1-5	73.05	82.32	85.05	85.84

5.2.3 Pitch tracking by clustering local fundamental frequency estimates

Multiple fundamental frequency (f0) estimation is one of the most important problems in music signal analysis and constitutes a fundamental step in several applications such as melody extraction, sound source identification and separation. We have shown that the Fan Chirp Transform (FChT) can be applied to polyphonic music analysis, and produces the pitch salience representation F0gram that provides a set of local fundamental frequency candidates together with a pitch change rate estimate for each of them.

To continue the analysis temporal integration of local pitch candidates has to be performed. There is a vast amount of research on pitch tracking in audio, often comprising an initial frame by frame f0 estimation followed by formation of pitch contours exploiting estimates continuity over time. Techniques such as dynamic programming, linear prediction, Hidden Markov Models, among many others (see [43] for a review), were applied to the temporal tracking.

The technique we discuss in this section for pitch contours formation does not involve a classical temporal tracking algorithm. Instead the pitch tracking is performed by unsupervised clustering of F0gram peaks. A spectral clustering method [44] is selected for this task as it imposes no assumption of convex clusters, thus being suitable for filiform shapes such as pitch contours. The pitch change rate estimates provided by the FChT analysis play an important role in the definition of similarity between pitch candidates. The clustering is carried out within overlapped observation windows corresponding to several signal frames. Then contours are formed by simply joining clusters that share elements. This short-term two-stage processing proved to be more robust than aiming a straightforward long-term clustering.

There are very few applications of spectral clustering for tracking a sound source. Blind one-microphone separation of two speakers is tackled in [45] as a segmentation of the spectrogram. A method is proposed to learn similarity matrices from labeled datasets. Several grouping cues are applied such as time–frequency continuity and harmonicity based. A simple multiple pitch estimation algorithm is part of the feature extraction. The mixing conditions are very restrictive (equal strength and no reverberation). Performance is assessed through a few separation experiments.

Clustering of spectral peaks is applied in [46], for partial tracking and source formation. Connecting peaks over time to form partials and grouping them to form sound sources is performed simultaneously. The problem is modeled as a weighted undirected graph where the nodes are the peaks of the magnitude spectrum. The edge weight between nodes is a function of frequency and amplitude proximity (temporal tracking) and a harmonicity measure (source formation). Clustering of peaks across frequency and time is carried out for windows of an integer number of frames ($\sim$ 150 ms) using a spectral clustering method. Clusters from different windows are not connected for temporal continuation. The two more compact clusters of each window are selected as the predominant sound source.

5.2.3.1 Pitch Salience Computation

We will use the FChT as a low level representation and exploit the chirp rate information, to develop the tracking technique. As we aforementioned, computing the FChT for consecutive short time signal frames a time–frequency representation in the form of a spectrogram can be built. For polyphonic music analysis,

several FChT instances with different α values can be computed. This produces a multidimensional representation made up of various time–frequency planes. For each f0 in the highest salience value is selected among the different available α values. In this way an F0gram is obtained, that shows the evolution of pitch for all the harmonic sounds in the signal.

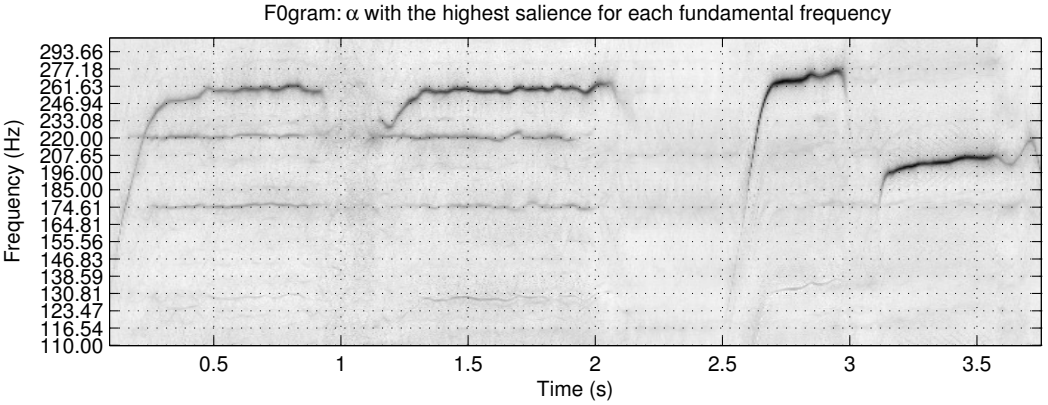


FIGURE 5.9: F0gram for a fragment of the audio file pop1, from the MIREX melody test set. It consist of three simultaneous singing voices followed by a single voice, with a rather soft accompaniment.

5.2.3.2 Spectral Clustering

The goal of clustering can be stated as dividing data points into groups such that points in the same cluster are similar and points in different clusters are dissimilar. An useful way of representing the data is in the form of a similarity graph, each vertex corresponding to a data point. Two vertices of the graph are connected if their similarity is above certain threshold, and the edge between them is weighted by their similarity value. In terms of the graph representation, the aim of clustering is to find a partition of the graph such that different groups are connected by very low weights whereas edges within a group have high weights.

The simplest way to construct a partition is to solve the mincut problem. Given a number of clusters k, it consist in finding a partition $A_1, \ldots, A_k$ that minimizes

$$\text{cut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i), \tag{5.2}$$

where $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$ is the sum of weights of vertices connecting partitions A and B, and $\bar{A}$ stands for the complement of A. This corresponds to find

a partition such that points in different clusters are dissimilar to each other. The problem with this approach is that it often separates one individual vertex from the rest of the graph. An effective way of avoiding too small clusters is to minimize the Ncut function,

$$\mathrm{Ncut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)}, \tag{5.3}$$

where $\mathrm{vol}(A) = \sum_{i \in A} d_i$ is the sum of the degree of vertices in A. The degree of a vertex is defined as $d_i = \sum_{j=1}^{n} w_{ij}$, so $\mathrm{vol}(A)$ measures the size of A in terms of the sum of weights of those edges attached to their vertices. The Ncut criterion minimizes the inter–cluster similarity (in the same way as mincut), but it also implements a maximization of the intra–cluster similarities. Notice that the intra–cluster similarity can be expressed as, $W(A, A) = \mathrm{vol}(A) - \mathrm{cut}(A, \bar{A})$ [47]. In this way the Ncut criterion implements both objectives: to minimize the inter–cluster similarity, if $\mathrm{cut}(A, \bar{A})$ is small, and to maximize the within cluster similarity, if $\mathrm{vol}(A)$ is large and $\mathrm{cut}(A, \bar{A})$ is small.

The mincut problem can be solved efficiently. However with the normalization term introduced by Ncut it becomes NP hard. Spectral clustering is a way to solve relaxed versions of this type of problems. Relaxing Ncut leads to the normalized spectral clustering algorithm. It can be shown [47] that finding a partition of a graph with n vertices into k clusters by minimizing Ncut, is equivalent to finding k indicator vectors $h_j = (h_{1j}, \ldots, h_{nj})^T$ with $j = 1, \ldots, k$ of the form, $h_{ij} = 1/\mathrm{vol}(A_j)$ if vertex $v_i \in A_j$ and zero otherwise. In this way, the elements of the indicator vectors point out to which cluster belongs each graph vertex. This problem is still NP hard, but can be relaxed by allowing the elements of the indicator vectors to take, instead of two discrete values, any arbitrary value in $\mathbb{R}$. The solution to this relaxed problem corresponds to the first k generalized eigenvectors of $(D - W)\, u = \lambda\, D\, u$, where D is an n by n diagonal matrix with the degrees of the graph vertices $d_1, \ldots, d_n$ on the diagonal, and $W = (w_{ij})_{i,j=1\ldots n}$ is the matrix of graph weights.

The vectors u of the solution are real-valued due to the relaxation and should be transformed to discrete indicator vectors to obtain a partition of the graph. To do this, each eigenvalue can be used in turn to bipartite the graph recursively by finding the splitting point such that Ncut is minimized [44]. However, this heuristic may be too simple in some cases and most spectral clustering algorithms consider the coordinates of the eigenvectors as points in $\mathbb{R}^k$ and cluster them

using an algorithm such as k-means [47]. The change of representation from the original data points to the eigenvector coordinates enhances the cluster structure of the data, so this last clustering step should be very simple if the original data contains well defined clusters. In the ideal case of completely separated clusters the eigenvectors are piecewise constant so all the points belonging to the same cluster are mapped to exactly the same point.

To check the normalized spectral clustering algorithm presented in [44], please refer to [47].

5.2.3.3 Pitch Contours Formation

In order to apply the spectral clustering algorithm to the formation of pitch contours several aspects must be defined. In particular, the construction of the graph involves deciding which vertices are connected. Then, a similarity function has to be designed such that it induces meaningful local neighbours. Besides, an effective strategy has to be adopted to estimate the number of clusters. In what follows, each of these issues are discussed and the proposed algorithm is described.

5.2.3.4 Graph construction

Constructing the similarity graph is not a trivial task and constitutes a key factor in spectral clustering performance. Different alternatives exist for the type of graph, such as k-nearest neighbor, ϵ-neighborhood or fully connected graphs, which behave rather differently. Unfortunately, barely any theoretical results are known to guide this choice and to select graph parameters [47]. A general criteria is that the resulting graph should be fully connected or at least should contain significantly fewer connected components than the clusters we want to detect. Otherwise, the algorithm will trivially return connected components as clusters.

To include information on temporal proximity a local fixed neighborhood is defined, such that f0 candidates at a certain time frame are connected only to candidates in their vicinity of a few frames (e.g. two neighbor frames on each side). In this way the graph is in principle fully connected, as can be seen in Figure 5.10, and resulting connected components are determined by similarity between vertices. Two candidates distant in time may nevertheless belong to the same cluster by their similarity to intermediate peaks. Note that in Figure 5.10 only one neighbor frame on each side is taken into account to link peaks. In this case, if

a peak is missing the given contour may be disconnected. For this reason, a local neighbourhood of two or three frames on each side is preferred. Similarity of not connected components is set to zero, so a sparse similarity matrix is obtained.

In addition, a contour should not contain more than one f0 candidate per frame. To favour this, candidates in the same frame are not connected. Specifying cannot-link constrains of this type is a common approach for semi-supervised clustering [48]. However, this not strictly prohibits two simultaneous peaks to be grouped in the same cluster if their similarity to neighbor candidates is high. For this reason, clusters should be further processed to detect this situation and select the most appropriate candidate in case of collisions.

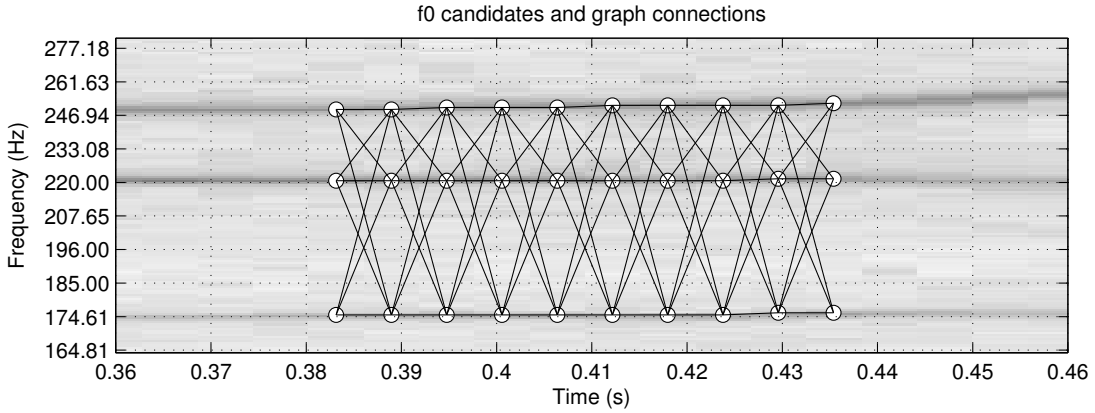


FIGURE 5.10: Graph connections considering only one neighbor frame on each side for an observation window of 10 frames. The resulting graph includes all the connections between consecutive frames.

5.2.3.5 Similarity measure

To define a similarity measure between F0gram peaks it is reasonable to base it on the assumption of slow variation of pitch contours in terms of fundamental frequency and salience (as defined in Section 5.2.3.1).

Fundamental frequency distance between two graph vertices v_i and v_j may be better expressed in a logarithmic scale, that is as a fraction of semitones. To do this, pitch value of a vertex is expressed as the corresponding index in the logarithmically spaced grid used for pitch salience computation[6]. Then, this can be converted to a similarity value $s_{f0}(v_i, v_j) \in (0, 1]$ using a Gaussian radial basis

[6]In which a 16th semitone division is used (192 points per octave).

function,

$$s_{f0}(v_i, v_j) = e^{-\frac{d^2_{f0}(v_i,v_j)}{\sigma^2_{f0}}} \tag{5.4}$$

where $d_{f0}(v_i, v_j) = |f0_i - f0_j|$ stands for pitch distance and σ_{f0} is a parameter that must be set which defines the width of local neighborhoods. In a similar way, a similarity function can be defined that accounts for salience proximity. To combine both similarity functions they can be multiplied, as in [46].

Although this approach was implemented and proved to work in several cases, the similarity measure has some shortcomings. Pitch based similarity is not able to discriminate contours that intersect. In this case, salience may be useful but it also has some drawbacks. For instance, points that are not so near in frequency and should be grouped apart, may be brought together by their salience similarity. This suggest the need for a more appropriate way of combining similarity values.

A significant performance improvement was obtained by combining the pitch value of the candidates and the chirp rates provided by the FChT. The chirp rate can be regarded as a local estimation of the pitch change rate. Thus, the pitch value of the next point in the contour can be predicted as $\overrightarrow{f0}_i^k = f0_i^k\,(1 + \alpha_i^k\,\Delta t)$, where $f0_i^k$ and α_i^k are the pitch and chirp rate values, i and k are the candidate and frame indexes respectively, and Δt is the time interval between consecutive signal frames. Figure 5.11 depicts most prominent f0 candidates and their predictions for a short region of the example. Note that there are some spurious peaks in the vicinity of a true pitch contour whose estimate lie close to a member of the contour and can lead to an incorrect grouping. A more robust similarity measure can be obtained by combining mutual predictions between pitch candidates. This is done by computing for each candidate also a backward prediction $\overleftarrow{f0}$ in the same way as before. Then, distance among two candidates v_i^k and v_j^{k+1} is obtained by averaging distances between their actual pitch values and their mutual predictions,

$$d_{f0}(v_i^k, v_j^{k+1}) = \frac{1}{2}\left[|f0_i^k - \overleftarrow{f0}_j^{k+1}| + |\overrightarrow{f0}_i^k - f0_j^{k+1}|\right] \tag{5.5}$$

Using this mutual distance measure the similarity function is defined as in Equation (5.4). Additionally, the same reasoning can be extended to compute forward and backward predictions for two or three consecutive frames. This similarity values are used as graph weights for candidates in their temporal proximity.

Still remains to set the value σ_{f0}, which plays the role of determining the actual value assigned to points in the vicinity and to outlying points. Self tunning sigma for each pair of data points was tested based on the distance to the k-th nearest

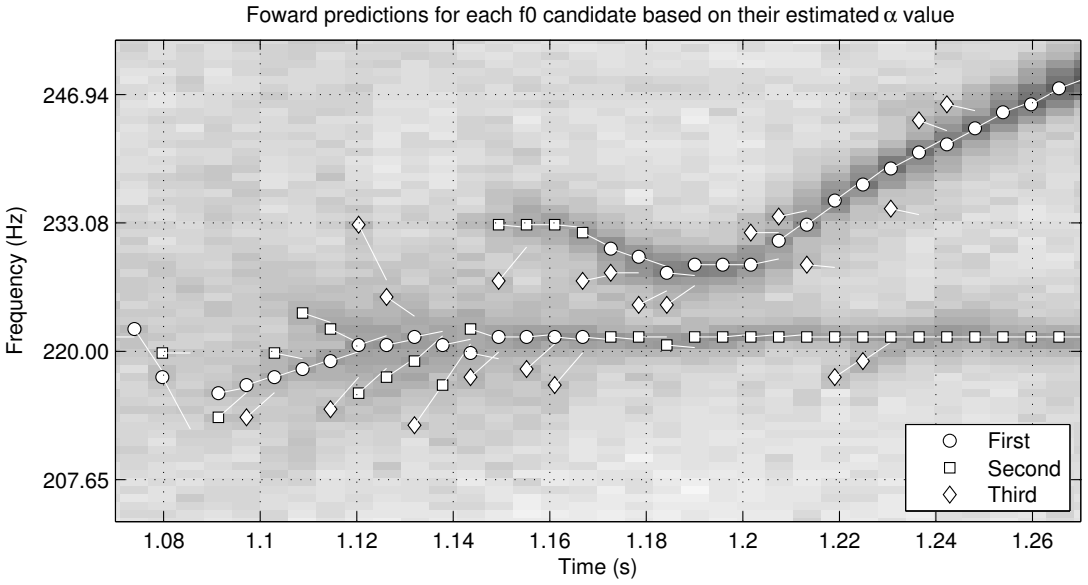


FIGURE 5.11: Forward predictions of the three most prominent f0 candidates for a short interval of the example. Although the clusters seem to emerge quite defined, spurious peaks may mislead the grouping. This can be improved if a backward prediction is also considered.

neighbor of each point, as proposed in [49]. This approach can handle cluster with different scales, but applied to this particular problem it frequently grouped noisy peaks far apart from each other. It turned out that, given the filiform shape of clusters that are to be detected, a fixed value for σ_{f0} was more effective. Since pitch predictions become less reliable as the time interval grows, a more restrictive value for σ_{f0} is used for measuring similarity to points at the second and third consecutive frame (reported results correspond to $\sigma^1_{f0} = 0.8$, $\sigma^2_{f0} = 0.4$).

Figures 5.12 and 5.13 show two different examples of the local clustering, which correspond to three well-defined clusters and two clusters with spurious peaks. An observation window of 10 signal frames is used and the three most prominent F0gram peaks are considered. A neighborhood of two frames on each side is used. Similarity matrix is sorted according to the detected clusters, producing a sparse band diagonal matrix, where clusters can be visually identified as continuous bands.

5.2.3.6 Number of clusters determination

Automatically determining the number of clusters is a difficult problem and several methods have been proposed for this task [48]. A method devised for spectral

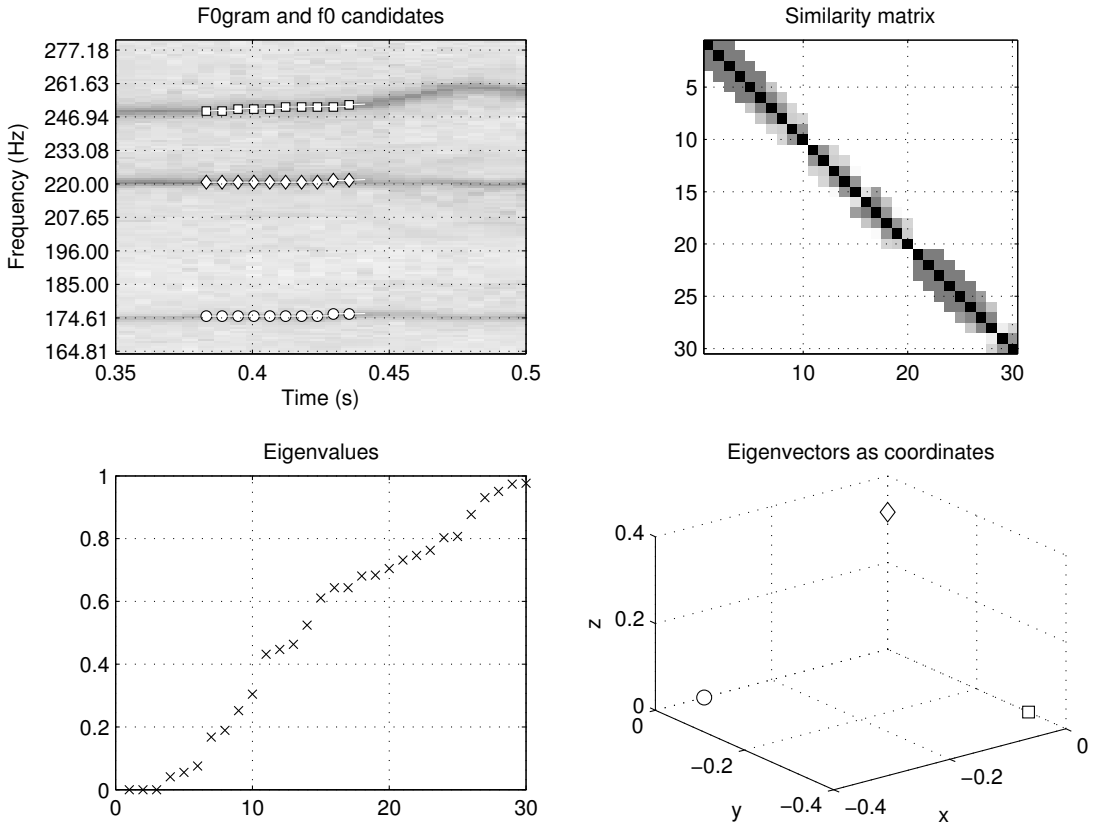


FIGURE 5.12: Local clustering for a short time interval of the audio example. Similarity matrix, eigenvalues and eigenvectors as coordinates are depicted. Three well-defined clusters can be identified in the data, as well as the corresponding bands in the similarity matrix. The multiplicity of eigenvalue zero coincides with the number of connected components. All members of a cluster are mapped to the same point in the transformed space.

clustering is the eigengap heuristic [47]. The goal is to choose the number k such that all eigenvalues $\lambda_1, \ldots, \lambda_k$ are very small, but λ_{k+1} is relatively large. Among the various justifications for this procedure, it can be noticed that in the ideal case of k completely disconnected components, the graph Laplacian $L = (D - W)$ has as many eigenvalues zero as there are connected components, and then there is a gap to the next eigenvalue.

5.2.3.7 Formation of pitch contours

The above described local clustering of f0 candidates has to be extended to form pitch contours. Increasing the length of the observation window showed not to be the most appropriate option. The complexity of the clustering is increased for longer windows, since a higher number of clusters inevitably arise mainly because

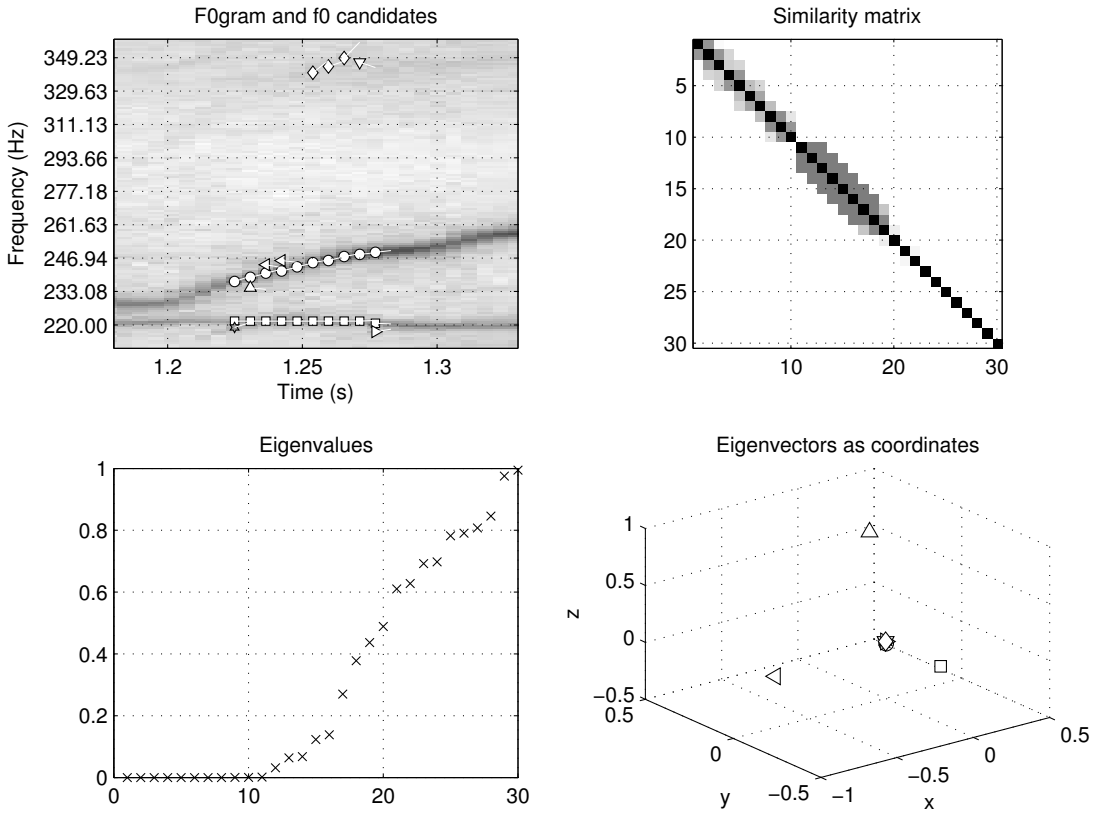


FIGURE 5.13: Local clustering example with two true pitch contours and several spurious peaks. The two corresponding bands in the similarity matrix can be appreciated. Multiplicity of eigenvalue zero not only indicates the relevant connected components but also isolated points. The true contours are correctly identified by the algorithm and spurious peaks tend to be isolated.

of spurious peaks. Additionally, computational burden grows exponentially with the number of graph vertices. Thus, an observation window of 10 signal frames was used in the reported simulations ($\sim$ 60 ms).

Neighboring clusters in time can be identified based on the similarity among their members. A straightforward way to to this is by performing local clustering on overlapped observation windows and then grouping clusters that share elements. Figure 5.14 shows the clustering obtained using half overlapped observation windows for the two previously introduced examples.

5.2.3.8 Results and discusion

The contours obtained by applying the clustering algorithm to the example audio excerpt are depicted in Figure 5.15. The three most prominent peaks of the F0gram are considered for pitch tracking. Several issues can be noted from these results.

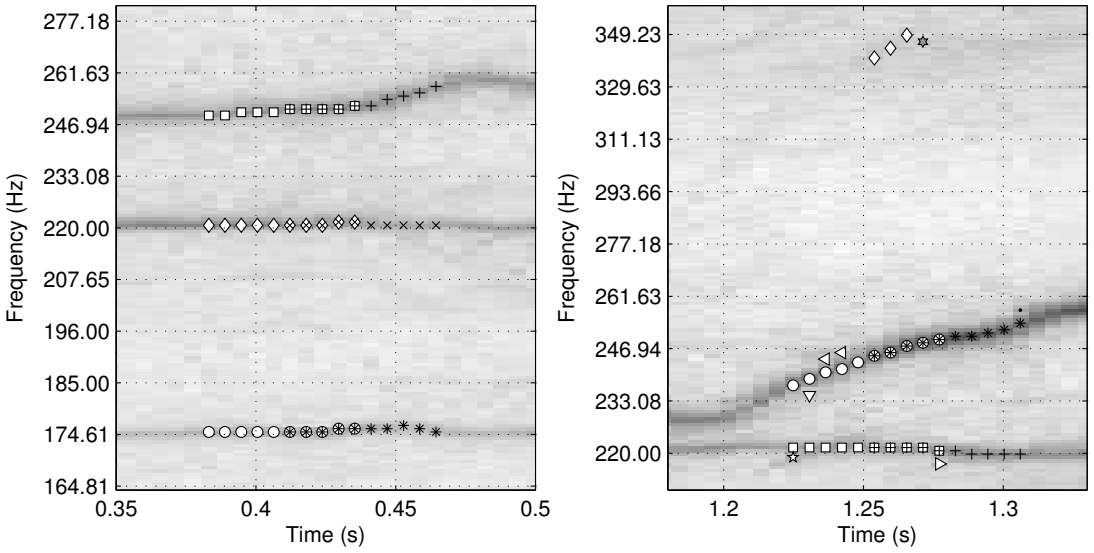


FIGURE 5.14: Examples of clustering using half overlapped observation windows. Clusters from different time windows are linked if they share elements. Each cluster is represented with a different marker. In both examples, pitch contours are correctly continued with this criterion since half of their members are shared. The pitch contours are correctly continued since several of their members are shared.

Firstly, the main contours present are correctly identified, without the appearance of spurious detections when no harmonic sound is present (e.g. around $t = 1.0\,s$). The example shows that many sound sources can be tracked simultaneously with this approach. No assumption is made on the number of simultaneous sources, which is only limited by the number of pitch candidates considered. The total number of contours and concurrent voices at each time interval is derived from the data.

It can also be seen that the third voice of the second note (approximately at $t = 1.0 - 2.0\,s$) is only partially identified by two discontinued portions. Since the low prominence of this contour some of the pitch candidates appear as secondary peaks of the more prominent sources. This situation can be improved by increasing the number of prominent peaks considered.

Apart from that, there are the three short length contours detected at interval $t = 2.1 - 2.5\,s$ that seem to be spurious. However, when carefully inspecting the audio file it turned out that they correspond to harmonic sounds from the accompaniment. Although this contours have a very low salience they are validated because of their structure. It depends on the particular problem where this

algorithm finds application if these contours may be better filtered out based on salience.

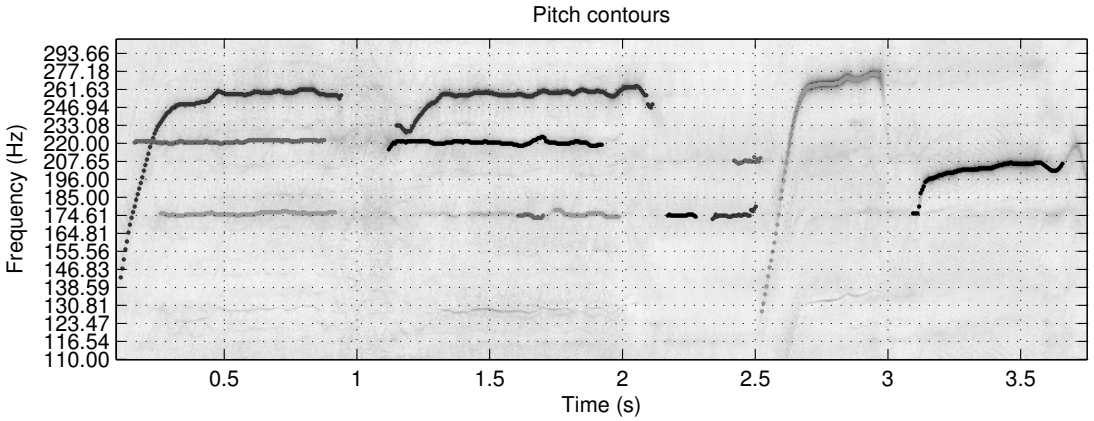


FIGURE 5.15: Pitch contours for the audio example obtained by considering the three most prominent F0gram peaks. Each contour is shown with a different shade of grey.

A melody detection evaluation was conducted following a procedure similar to the one applied in [50]. The vocal files of the 2004-2005 MIREX melody extraction test set were considered, which is a publicly labeled database available from http://www.music-ir.org/mirex/. It comprises 21 music excerpts for a total duration of 8 minutes.

The three most prominent F0gram peaks were selected as pitch candidates to form contours using the herein described algorithm. All the identified pitch contours were considered as main melody candidates and the ones that better match the labels were used to asses performance. Only those frames for which the melody was present according to the labels were taken into account to compute the evaluation measure according to,

$$\text{score}(f_0) = \min\{1, \max\{0, (\text{tol}_\text{M} - \Delta f_0)/(\text{tol}_\text{M} - \text{tol}_\text{m})\}\}$$

where $\Delta f_0 = 100|f_0 - f_0^{gt}|/f_0^{gt}$ is the relative error between the pitch contour value and the ground truth, and the tolerances tol_M and tol_m correspond to 3% and 1% respectively. This represents a strict soft thresholding.

The performance obtained in this way is compared to an equivalent evaluation that considers F0gram peaks as main melody estimates without performing any type of grouping into contours (as reported in [50]). Grouping the F0gram peaks into contours involves the determination of where does a contour starts and when does

it ends, necessarily leaving some time intervals without melody estimation. This is avoided when isolated F0gram peaks are considered as main melody estimates, since for every melody labeled frame there is always a pitch estimation. Therefore, this performance measure can be considered as a best possible reference.

Results of the evaluation are presented in Table 5.2. Two different values are reported for the pitch contours formation corresponding to a single run of the k-means algorithm and 10 repetitions. When the clusters in the transformed space are not well defined the k-means algorithm can get stuck in a local minima. This can be improved if several executions are performed but with different set of initial cluster centroid positions and the best performing solution is returned (i.e. lowest centroid distances). It can be noticed that the k-means repetition consistently gives a slight performance increase.

In addition, precision and recall values are reported. Precision is computed as the mean score value of the estimations within the 3% threshold. Remaining frames are considered not recalled items, as well as melody labeled frames for which there is no pitch contour.

When visually inspecting the results for individual files it turned out that most melody labeled regions for which there were no estimated contours correspond to low salience portions of the F0gram (for instance, when a note vanishes). It seems that labels are produced from monophonic files containing only the vocal melody and when mixed into a polyphonic track some regions are masked by the accompaniment. Figure 5.16 shows a detail of the current example where this situation can be appreciated. In order to take this into account the evaluation was repeated but ignoring low prominent melody frames. To do this a salience estimation was obtained for each labeled frame by interpolating the F0gram values. Then a global threshold was applied to discard those frames whose salience was below 30% of the F0gram maximum value (26% of the total frames).

The performance of the pitch contours formation by itself is quite encouraging. However, it decreases considerably compared to the values obtained before grouping F0gram peaks. The gap is reduced by restricting the evaluation to the most prominent peaks, which seems to confirm that low salience regions are troublesome for the algorithm. Visually inspecting the estimations for individual files gives the idea that most pitch contours are correctly identified. However, the evaluation results indicate the algorithm seems not to take full advantage of the information given by the F0gram peaks. Blindly relying on estimated α values no matter

TABLE 5.2: Results for the melody detection evaluation. The pitch contours are obtained from the three most prominent f0 candidates. An evaluation using F0gram peaks (1st to 3rd) without tracking is also reported.

F0gram peaks	no salience threshold score	30% salience threshold score
1	83.38	97.22
1-2	88.24	99.20
1-3	90.33	99.61
Pitch contours	**no salience threshold score**	**30% salience threshold score**
1 k-means	81.99	96.69
10 k-means	83.21	97.20
frames	100%	74%

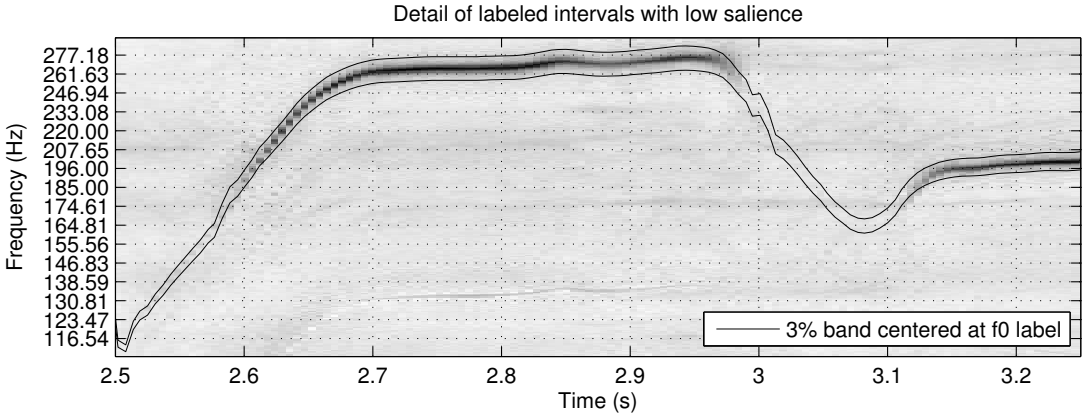


FIGURE 5.16: Some melody labeled regions of the example exhibit a very low salience (2.5-2.6 and 3.0-3.1 s).

their corresponding salience is probably the most important shortcoming of the proposed algorithm.

5.2.3.9 Conclusions

A way of performing pitch tracking by means of clustering local f0 candidates is described. This makes use of the Fan Chirp Transform which can produce precise representation of non stationary sound sources like singing voice. The tracking is based on clustering techniques and makes use of the salience, pitch and pitch change rate that the FChT provides.

The grouping is performed by applying a Spectral Clustering method since it can handle filiform shapes such as pitch contours. The similarity measure proposed takes advantage of the pitch change rate estimate provided by the FChT based

F0gram which is information that usually is not available in other low level representations. The determination of the number of clusters is done in an iterative approach, where the number of connected components is taken as an initial estimate and not compact enough clusters are further divided into an increasing number of groups. This strategy tends to isolate each spurious peak in a single cluster, what in turn favours to ignore them in the formation of pitch contours. Clustering is carried out for overlapped observation windows of a few hundred milliseconds and clusters from different time windows are linked if they share elements. In this way, groups that exhibit a coherent geometric structure emerge as pitch contours while the others are discarded.

Results of a melody detection evaluation indicate that the introduced technique is promising for pitch tracking and can effectively distinguish most singing voice pitch contours. There is some room for improvement. In particular, other sources of information can be included in the similarity measure in order to take full advantage of the local pitch candidates. The estimation of the pitch change rate is less reliable for low salience peaks. This could be taken into account when computing similarity, for example by adjusting the σ_{f0} value in accordance with the salience of the candidate.

Part II

Audio Source Separation Techniques and Applications

Chapter 6

Source Separation based on Gaussian Mixture Models

In this Chapter we will focus in an application where the fundamental frequency of the sources is assumed to be known. Sparse representation techniques combined with Gaussian Mixture Models were applied in a Score-Informed Source Separation problem.

The description of the techniques used herein are based in the publication [25], in collaboration with Pablo Sprechmann and Guillermo Sapiro. The Chapter reproduces some the article passages, as well as includes some modifications or additions in order to contribute to the structure of this document.

6.1 Introduction

Since musical scores are easily available and provide fundamental information about the musical piece, in this work we tackle the problem of *score-informed* Single Channel Source Separation (SCSS) in musical pieces [51–53]. The information extracted from the scores is used as prior information to initialize and guide our algorithm. The Fan Chirp Transform was not included in the solution to this problem, as we will assume that the instruments in the mix do not have big pitch fluctuations, and hence there would not be much improvement using it.

The decomposition of time–frequency representations, such as the power spectrogram, in terms of elementary atoms of a dictionary has become a popular tool in audio processing. In particular, non-negative matrix factorization (NMF), [54],

leads to good results in different applications. Single channel source separation via NMF is carried out by decomposing the magnitude spectrogram of the mixture signal and then performing reconstructions of groups corresponding to each single source. In the fully unsupervised setting, these methods brake down when the sources have a large time-overlap in the track. A considerable amount of work has been dedicated to add constraints to the factorization in order to include prior information guiding the challenging decomposition.

When NMF is applied to quasi-harmonic instrument sounds, the elementary components that are redundant throughout the piece will hopefully represent musical notes and thus have a harmonic structure. However, this cannot be guaranteed. Recent methods have proposed constraining the atoms to have particular designs following prior information about the signal in order to obtain physically meaningful atoms, and in particular, to mimic the harmonic structure present in the spectrograms of musical instruments [53, 55–57].

The variability in the spectrum of a musical instrument sound has two main components: the variation of the fundamental frequency and the changes in its spectral envelope. Standard NMF has shown good results when the characteristics of the sounds are stationary. In other words, NMF relies in the frame-to-frame redundancy. Slight changes in the fundamental frequency with constant spectral envelope produce severe changes in the spectrogram. The same happens when changes in the spectral envelope occur with a fixed pitch. Classical NMF will likely need several dictionary atoms to account for this variability, while the nature of these changes is rather simple.

We propose a simple model for representing the spectrogram magnitude of musical instruments that decouples between the information of pitch and spectral envelope. This allows to represent efficiently a great deal of variability using very simple models for each component. Specifically, let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N] \in \mathbb{R}^{F \times N}$ be the power spectrogram of a signal containing, for now, an isolated instrument. We can decompose this as

$$\mathbf{V} \approx \mathbf{H} \bullet \mathbf{E}, \tag{6.1}$$

where $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_N] \in \mathbb{R}^{F \times N}$ is a non-negative matrix modeling the spectral envelope and its evolution in time, and $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N] \in \mathbb{R}^{F \times N}$ is a matrix with entries in the $[0, 1]$ interval enforcing the harmonic structure through an element-wise multiplication $\bullet$. With this representation, changes in pitch are captured in

H, while changes in the timbre appear in **E**. The space of the spectral envelope is simpler and can be accurately represented via Gaussian modeling.

The proposed method is particularly well suited for score-informed SCSS, since the pitch of each source is (approximately) known beforehand, having a very good guess of the shape of **H** in (6.1). Given **H**, finding **E** can be cast as an inverse problem. This formulation is inspired in part by the excellent results reported by [58] for a number of inverse problems in image processing.

Following [58], the Gaussian parameters and the signal representation are simultaneously estimated via an efficient MAP-EM (maximum a posteriori expectation-maximization) algorithm. Once the decomposition is obtained, we can construct a time–frequency mask source, recovering each source from the mixture by Wiener filtering.

In [53] the authors proposed a method based on NMF that also uses parametric templates to represent the harmonic structure of notes. Timbre is represented by the estimation of the amplitude of partials. The main difference with the proposed approach is that, for each instrument, all the notes are assumed to share the same fixed relative harmonics amplitudes. The Gaussian modeling allows a more flexible representation capturing the frequency-dependent resonance of the instruments and their variability.

In Section 6.2 we present the proposed audio (instruments) signal modeling and describe the algorithm for performing score-informed SCSS. Section 6.5 discusses connections with PCA and factorization methods. In Section 6.6 we evaluate the method with synthetic and real data.

6.2 Single Signal Model

We assume that each source is stationary within a frame. This means that for the i-th frame, the quasi-harmonic part of the signal can be considered as harmonic with a fundamental frequency f_i. In the Fourier domain, most of the energy of $\mathbf{v_i}$ is concentrated in bins corresponding to frequencies of the form kf_i, with $k \in \mathbb{Z}$. Rewriting (6.1) in a frame basis we obtain

$$\mathbf{v}_i = \mathbf{h}_i \bullet \mathbf{e}_i + \mathbf{w}_i, \quad \text{for} \quad i = 1, \ldots, N,$$

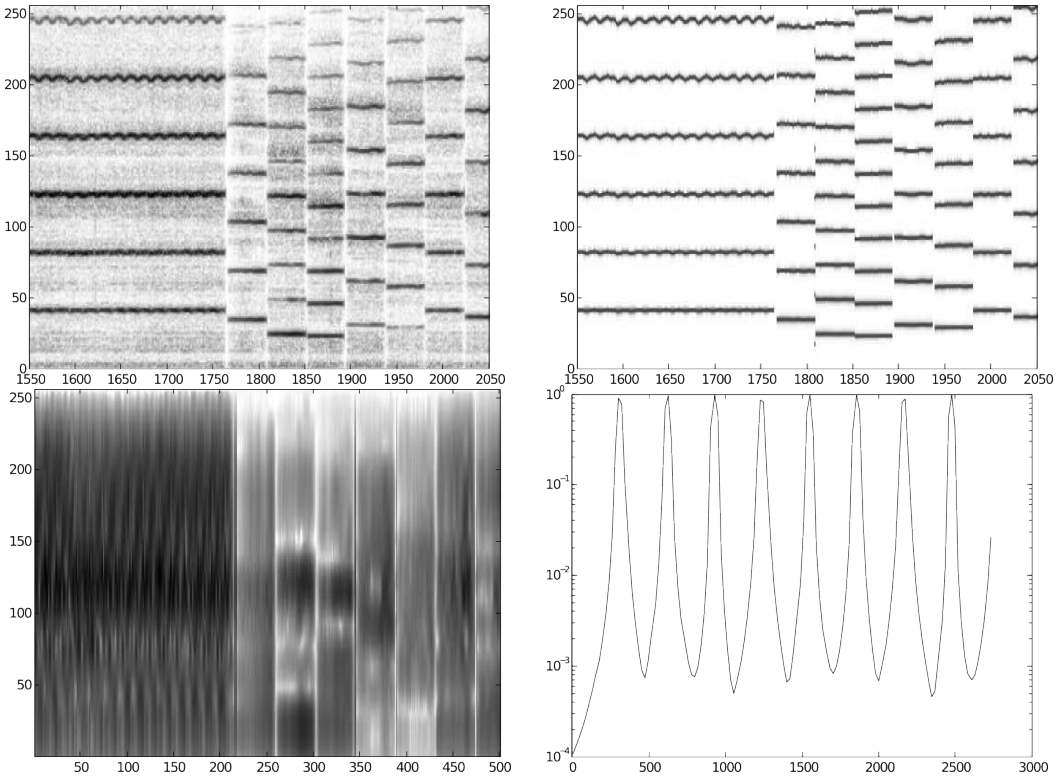


FIGURE 6.1: In this figure we show, respectively, an example of a spectrogram $\mathbf{V}$, the corresponding matrices $\mathbf{H}$ and $\mathbf{E}$, and a comb filter $\mathbf{h}_i$.

where $\mathbf{h}_i$ is the power spectrum of a linear filter that enforces an harmonic constraint on the representation, $\mathbf{e}_i$ is the envelope of the spectral content, and $\mathbf{w}_i$ is a representation error. We consider $\mathbf{h}_i$ to be the spectral response of a comb filter with unit amplitude and parametrized by its fundamental frequency, $\mathbf{h}_i = \mathbf{h}(f_i)$. The point wise multiplication $\mathbf{h}_i \bullet \mathbf{e}_i$ can be seen as the aplication of a comb filter to $\mathbf{e}_i$. In every frame, $\mathbf{e}_i$ is assumed to be drawn from a Gaussian distribution with mean μ and covariance Σ, to be learned. The representation error is assumed to be also Gaussian with zero mean and known or estimated signal-independent isotropic covariance $\sigma^2 I_d$. See Figure 6.1 for an illustration of this model.

The available score provides a set of possible values for the fundamental frequency, $\mathbf{f}_0 = [f_{01}, \ldots, f_{0N}]$, which are a very good approximation of the true values of $\mathbf{f} = [f_1, \ldots, f_N]$. However, they cannot be assumed to be identical. A canonical example of such a situation is the vibrato, where the fundamental frequency slightly oscillates around a specific note. To deal with this variability we model the fundamental frequency as a time changing (real valued) Gaussian distribution

centered at the fundamental frequency given by the score and with variance σ_0^2.[1]

6.3 Mixed Signal Model

Let's now assume that the signal is a mixture of c quasi-harmonic instruments. We want to decompose its power spectrum as

$$\mathbf{V} = \sum_{j=1}^{c} \mathbf{H}_j \bullet \mathbf{E}_j + \mathbf{W}. \tag{6.2}$$

The pitch and spectral envelope for the j-th instrument are modeled by the matrices $\mathbf{H}_j = [\mathbf{h}_{j1}, \ldots, \mathbf{h}_{jN}]$ and $\mathbf{E}_j = [\mathbf{e}_{j1}, \ldots, \mathbf{e}_{jN}]$ respectively, following the model presented in 6.2. As with NMF, the model estimation and the signal coding are done simultaneously:

- Estimating the Gaussian parameters $\mathcal{G} = \{(\mu_j, \Sigma_j)\}_{1 \leq j \leq c}$ for each instrument.
- Estimating the set of envelopes, $\{\mathbf{E}_j\}_{1 \leq j \leq c}$, and the real fundamental frequencies, $\{\mathbf{f}_j\}_{1 \leq j \leq c}$, from the spectrum $\mathbf{V}$ given, via the score, the set of corresponding fundamental frequencies,$\{\mathbf{f}_{0j}\}_{1 \leq j \leq c}$, and the Gaussian distributions for each instrument, $\mathcal{G}$.

To solve this non-convex problem we use an adaptation of the efficient MAP-EM algorithm presented for image processing in [58].

We could consider using a GMM to model the spectral content of the instruments instead of a single Gaussian distribution. This could help for example when modeling instruments with large timbre variability, i.e., plucked string sounds or singing voice.

6.4 Computational Algorithm

The MAP-EM algorithm is an iterative procedure that alternates between two steps. An *E-step* that estimates the spectral content of the sources assuming that

[1] In all our experiments we considered σ_0 to be 1% of the value of f_{0i}. However, σ_0^2 might be instrument dependent, since instruments such as the cello are normally played with vibrato, while others have a more constant pitch.

the Gaussian parameters are known, and an *M-step* that reciprocally estimates the Gaussian parameters for each source while assuming that the spectral envelopes are known. To simplify the notation and without loss of generality, the Gaussians are assumed to have zero mean, since they can be centered with respect to the estimated mean. We use the available score to produce a good initial condition for the algorithm (see Section 6.4.1). For more detail on how to compute the E-Step, and M-Step, please refer to [25]

6.4.1 Initialization

The initialization process for the MAP-EM goes as follows. First, the complete score gets synthesized producing an isolated track for each source. Then, each of these synthetic tracks is used to learn the Gaussian parameters for each instrument. In all the cases we assume that the signals are perfectly aligned to the available scores. The alignment problem can be solved automatically using dynamic time warping, see [52] and references therein.

6.5 Connections with PCA and NMF

In this section we first present an interpretation of the proposed method that links it with structured principal component analysis (PCA). Then we discuss its relations with the NMF.

6.5.1 Structured Estimation in PCA Bases

Given a set of signals $\{\mathbf{e}_{ji}\}_{1\leq i\leq N}$, the PCA basis are defined as the matrix $\mathbf{B}_j = [\mathbf{b}_{j1}, \ldots, \mathbf{b}_{jF}]$ that diagonalizes the corresponding covariance matrix, $\mathbf{\Sigma}_j = \mathbf{B}_j^T \mathbf{S}_j \mathbf{B}_j$, where $\mathbf{S}_j = \text{diag}(\lambda_1^j, \ldots, \lambda_F^j)$ is a diagonal matrix, whose diagonal elements are the sorted eigenvalues $\lambda_1^j \geq \lambda_2^j \geq \ldots \geq \lambda_F^j \geq 0$. The columns of $\mathbf{B}_j$ are orthonormal and represent the principal directions of variation of $\{\mathbf{e}_{ji}\}_{1\leq i\leq N}$. The magnitude of the eigenvalues measure the energy of the variation in the corresponding directions.

Working in the PCA basis rather than the canonical one, $\mathbf{a}_{ji} = \mathbf{B}_j^T \mathbf{e}_{ji}$, allows to significantly reduce the dimensionality of the data in a meaningful way. When representing the timbre of the instruments we can verify that they are highly

compressible: the first few eigenvalues of the covariance matrices concentrate most of the total energy. We can then write our model stated in (6.2) as

$$\mathbf{V} = \sum_{j=1}^{c} \mathbf{H}_j \bullet \mathbf{B}_j \mathbf{A}_j + \mathbf{W} \approx \sum_{j=1}^{c} \mathbf{H}_j \bullet \hat{\mathbf{B}}_j \mathbf{A}_j + \mathbf{W}, \tag{6.3}$$

where the matrices $\hat{\mathbf{B}}_j = [\mathbf{b}_{j1}, \ldots, \mathbf{b}_{jk}]$ conserve only the first $k \ll F$ principal directions. The MAP estimate can be computed as (see also [58])

$$\begin{aligned} \{\tilde{\mathbf{a}}_{ji}, \tilde{f}_{ji}\}_{1 \leq j \leq c} \quad &= \underset{\mathbf{a}_{ji}, f_{ji}}{\operatorname{argmin}} \|\mathbf{v}_i - \sum_{r=1}^{c} \mathbf{U}_{ri} \hat{\mathbf{B}}_j \mathbf{a}_{ri}\|^2 + \\ &\sigma^2 \sum_{r=1}^{c} \sum_{p=1}^{k} \frac{|a_{ri}[p]|^2}{\lambda_r^p} + \frac{\sigma^2}{\sigma_0^2} \sum_{r=1}^{c} |f_{ri} - f_{0ri}|^2. \end{aligned} \tag{6.4}$$

Inside each PCA basis, the atoms are pre-ordered by their corresponding eigenvalues. The weighting term in (6.4) privileges the coefficients corresponding to the principal directions with larger energy. This representation stabilizes the decomposition, which is crucial in these type of source separation ill-posed problems. We used $k = 25$ in all the experiments.

6.5.2 Links with NMF

In standard NMF, the spectrogram of the mixture signal is decomposed as the product of two non-negative matrices, $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}^{F \times Q}$, $\mathbf{H} \in \mathbb{R}^{Q \times N}$, and $Q \ll F, N$. The matrix $\mathbf{W}$ is the dictionary and each column represents an atom. The matrix $\mathbf{H}$ codes the activation of each atom in the dictionary throughout the frames. This representation is a low rank linear approximation of $\mathbf{V}$. Small variations in the pitch as the ones encountered in a vibrato for example, significantly increase the rank of $\mathbf{V}$, forcing to increase the number of atoms, Q. In SCSS this is highly undesirable, the models need to be as stable as possible in order to make sure that we can properly identify and reconstruct the multiple sources. A possible solution to address this problem is to add a sparsity constraint, so that only a few atoms of $\mathbf{W}$ are used for representing each frame [59].

In contrast to NMF, our proposed model is non-linear. We propose to use a linear model only to represent the timbre of each instrument, while using the set of parametric filters to model the harmonic constraint, as shown in (6.3). The proposed model is clearly less general than NMF since it is restricted to quasi-harmonic signals.

TABLE 6.1: Results with synthesis method M1 as testing signals.

	PLCA train M1	PLCA train M2	PDA	sPCA train M1	sPCA train M2	Oracle
SIR (dB)	22.8	14.2	20.2	17.9	17.8	20.5
SAR (dB)	11.5	6.3	7.7	11.0	10.9	13.6
SDR (dB)	11.1	4.8	7.2	10.8	10.0	12.7

TABLE 6.2: Results with synthesis method M2 as testing signals.

	PLCA train M1	PLCA train M2	PDA	sPCA train M1	sPCA train M2	Oracle
SIR (dB)	12.4	20.1	12.6	16.2	16.5	19.6
SAR (dB)	4.5	10.6	3.3	8.5	8.9	12.3
SDR (dB)	3.1	10.1	2.1	7.7	8.1	11.5

6.6 Numerical Experiments

In this section we evaluate the performance of the proposed method using real and synthetic data. The performance of the instruments separation methods is evaluated in terms of the standard measures: Signal to Interference Ratio (SIR), Signal to Artifact Ratio (SAR) and Signal to Distortion Ratio [60].[2] More information on these measures can be consulted in [60].

6.6.1 Synthetic Data

We used the publicly available database described in [53].[3] It contains 12 different string quartets (two violins, viola, and cello) by Bach, Beethoven and Boccherini. These pieces render important characteristics of real musical streams such as the overlap of the sources in the time–frequency domain. For each piece, the database provides a MIDI file containing the scores of the first 30 seconds of each piece, and two synthesized wave files for every individual instrument. The mixture signals are obtained by summing the individual tracks. The wave files are synthesized with one of two different methods. We will refer to these methods as $M1$ and $M2$ (see [53] for a detailed description). The signals synthesized with the different methods present distinct characteristics. For instance, they have different vibratos and decay times. Also, method $M2$ includes a reverberation effect present in the sound-font, while $M1$ does not.

[2] We used the BSS_EVAL toolbox [60].

[3] Available at `http://perso.enst.fr/hennequi/database.zip`.

	sPCA train SD	sPCA train RD	Oracle
SIR (dB)	17.4	17.5	18.2
SAR (dB)	11.1	11.2	13.2
SDR (dB)	10.1	10.6	11.7

TABLE 6.3: Results with MIREX 2007 MultiF0 database. Training using synthetic data (SD), real data from the database (RD), and (Oracle) as in 6.6.1.

Tables 6.1 and 6.2 compare our (sPCA) against the ones produced by [53] and a method based on PLCA [52] (refered to as PDAand PLCA respectively).[4] We report the results obtained by the proposed algorithm and PLCA using both the testing and training signals to train the models. The proposed algorithm is less sensitive to the initialization than the PLCA-based, therefore has better generalization properties. We also report the results obtained using the true isolated tracks to generate the Wiener masks. We refer to this as the Oracle method.

6.6.2 Real Data

We used the *Development Set for MIREX 2007 MultiF0 Estimation Tracking Task*.[5] It contains a 52 second long musical piece played by different wind instruments. The separated tracks for each instrument are available and the mixture is done by summing them. Table 6.3 shows the performance ratio obtained for a mixture of 4 instruments (clarinet, flute, horn and oboe) with different training conditions. The obtained results show that each signal is well captured.

6.7 Conclusions

A new method to represent quasi-harmonic audio signals that can be used to address audio source separation problems is described. The method is in particular applied to the score-informed source separation of musical mixtures. The method models the pitch and envelope of a sound source independently. This allows the modeling of signals that correspond to combinations of pitches and envelopes not previously observed. Moreover, it allows to easily incorporate meta-data such as the musical score indicating the signal to be represented. In this way, two or

[4] The results for PDAand PLCA were copied from [53].

[5] `http://www.music-ir.org/mirex/wiki/`

more musical instruments with the same characteristics can be separated. The characterized set of signals can be extended to incorporate non-harmonic sounds by defining filters $\mathbf{h_i}$ with the appropriate masking of spectral components. The method has been evaluated in the score-informed SCSS problem with synthetic and real data showing a performance comparable with those reported in the literature.

Chapter 7

Audio Source Separation based on the FChT

In this Chapter we describe the separation of harmonic sources from a mix based on the FChT. The separation relies on the fact that harmonic sources are naturally represented in a sparse representation if we correctly detect the chirp rate α making simple the isolation of the energy of one of these sources. The separation is obtained by inverting the steps of the FChT in order to re-synthesize the original isolated source.

7.1 Separation of harmonic sound sources

One of the applications in which the FChT properties become more advantageous is in the separation of sound sources from an audio mixture. One approach followed in this problem is to estimate the signal' spectrum and to select and extract the spectral peaks corresponding to a certain source. In this way, the waveform of the sound source is synthesized by the inverse Fourier Transform of the detected spectral components. The problem in its most general form has several difficulties, one of the most critical being the frequency resolution of non-stationary harmonic sources, when the fundamental frequency varies within the analysis window. In this case, the spectral peaks have a large bandwidth, which makes it difficult to discriminate the energy of the harmonics of the source to be detected from the remaining spectral components of the mixture.

In this context, the advantage of the FChT becomes clear. If the chirp rate matches the pitch change rate of a given source, its harmonics exhibit a minimum bandwidth. Thus, the automatic source detection is simplified, since the energy is much concentrated. Furthermore, the synthesized spectrum of the detected source is less contaminated by interference from other sources, given that a small number of frequency bins has to be preserved for each harmonic peak in a discrete Fourier Transform representation.

In order to isolate a source from a mix, the first step is to determine at each frame n the chirp rate $\alpha[n]$ and fundamental frequency $f_0[n]$ of the harmonic source. This can be done for example by means of the F0gram, as described in Section 3.3.4, but any other method to estimate the fundamental frequency variation within the analysis window time is also valid. The separation will be based on the transform calculated only for the selected chirp rate $\alpha[n]$. Almost all the energy of the source is concentrated in the bins that correspond to narrowbands centered at multiples of $f_0[n]$. The information that represents that source can be calculated by applying a mask that keeps the values on the corresponding bins to those bands and makes the rest null.

The following step is to combine the information from consecutive frames to reconstruct the signal. This process can be done with the Ovelapp-add technique, but taking into account that the sum of the overlapping windows is not constant nor known a priori. In the next section we describe how the inversion of a single frame is computed. Later an Overlap-add method is described to combine the signals form sequences of frames. Finally, the separation of a source is shown for a couple of examples, one synthetic and the other a real-world mix.

7.1.1 Inversion of a single frame

The calculation of the FChT involves the sequential application of a series of simple steps, as the non uniform resampling in time, windowing and the Fourier Transform. A straightforward starting point to analyse the reconstruction of a signal is to invert each of these steps in the opposite order as they were applied. We start the analysis with the inversion of the transform of a single frame to reconstruct the signal around the corresponding time of that frame. We will later discuss how to combine the information of several overlapping time frames to reconstruct the whole signal.

The expression of the FChT transform stated in continuous time is

$$\mathcal{F}\{w(t).x(\phi(t))\} \tag{7.1}$$

Where $w(t)$ is the applied window and $\phi(t)$ is the time warping function, see Equation 7.1. We first consider the inversion in time within the support of the window $w(t)$, as it does not make sense to recover values outside of this support. We will assume for the analysis of a individual frame inversion, that $w(t)$ does not take null values within the defined support. We will make the analysis of the inversion for the continuous time expression and point out the differences with the discrete time implementation.

The first step is the inversion of the Fourier Transform, which is invertible in both the continuous time and discrete time versions.

The next step is the application of the analysis window. Under the hypothesis we set, the original signal can be obtained by simply multiplying by the inverse value of the window. As we assumed that the window does not take null values at the considered times, the process is invertible. This limitation can be lifted if the information of contiguous frames is used together and the reconstruction values are obtained as a weighted sum of the contribution of each frame in a Overlap-Add fashion, where the sum of the weights cannot be null.

Finally, the time warping has to be undone. As the warping time function $\phi(t)$ we consider is monotonous, the process is invertible. In the discrete time implementation, the warping is implemented by a non uniform resampling of the signal. While there are no theoretical results on the invertibility of this process, excellent results are obtained if the resampling follows locally the Nyquist criteria. In other words, this means that the instantaneous sampling frequency is above two times the bandwidth of the signal, locally, at all times. Another consideration is the interpolation used in the resampling. The quality of the results is influenced by the interpolation method used, where optimal results are obtained using a band-limited interpolation. The results for a good interpolation method show that the residuals in the inversion are negligible compared to typical signal levels.

All this analysis was done for a given frame, and a given α. If we consider the full STFChT, we should merge the results from different time frames and different α's.

7.1.2 Overlap-add Integration

When the sum of the windows in the Overlap-add method is constant, the windowing process doesn't have to be corrected, as the contribution of consecutive frames perfectly reconstructs the original signal level.

In order to follow a similar reconstruction method we observe that the windowing process can be interchanged with the time warping. As they do not commute, some modifications have to be considered to obtain the same result. If the order of the warping and the window is changed, the window $w'(t)$ has to be such that after the warping it becomes $w(t)$, the actual analysis window. The window w' can be calculated from w applying the inverse time warping deformation.

$$w'(t) = w(\phi^{-1}(t))$$

So that:

$$w(t).x(\phi(t)) = [w(\phi^{-1}(t'))x(t')]|_{t'=\phi(t)}$$

The window $w'(t)$ will be different for each different warping, namely $w'_\alpha(t)$, so the windows in the unwarped original time depend on the chirp rate at each frame. Figure 7.1 shows an example of three different windows that correspond to different chirp rates. Considering a $w'(t)$ that is constant for different chirp rates would give different analysis windows in the warped time, not obtaining the desired properties of the analysis windows that offer windows like Hanning, Blackman, etc. As the chirp rate at each frame depends on the signal, there is no possible choice of the overlapping or the analysis window $w(t)$ that fulfills the constant overlap-add constraint.

The reconstruction not considering the weight of the windows, is calculated by summing the inverse of the Fourier Transform and warping of each frame n obtaining a signal $x_r(t)$. The window is not undone for each frame but for the whole signal. This can be written in the continuous time domain as:

$$x'_r(t) = \sum_n \mathcal{F}^{-1}\{X(f)\}|_{(\phi^{-1}(t)-nT_h)}$$

In order to correct the effect of the windows in the reconstruction of the signal, a straightforward approach is to consider the value of the signal as a weighted mean of the contributions of different frames. To do this, the signal has to be divided at

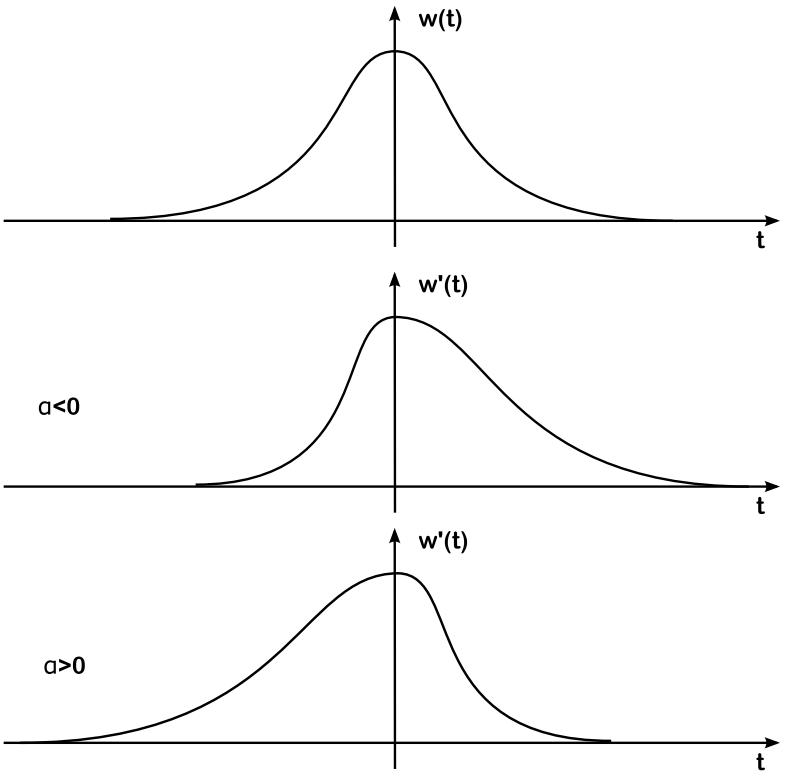


FIGURE 7.1: A window in the warped time and correspondent windows in the unwarped for positive and negative chirp rates.

each time by the value of the sum of the values of all the windows $w'_{\alpha[n]}(t - nT_h)$, where $\alpha[n]$ is the selected chirp rate for the frame n, and T_h is the hop time.

The correction equation of the overlapped windows weights in the reconstruction becomes:

$$x_r(t) = \frac{x'_r(t)}{\sum_n w'_{\alpha[n]}(t - nT_h)}$$

7.2 FChT for Audio Source Separation

In audio source separation it is usual practice to design algorithms that use the STFT as a low-level representation and produce a Modified STFT that represents the result. A typical way of calculating this MSTFT is usually by a Wiener filter, which takes as an input a mask that indicates how much of each component of the original STFT is part of the objective signal. As the MSTFT may not have consistent values that correspond to a true STFT of a piece of audio, different methods

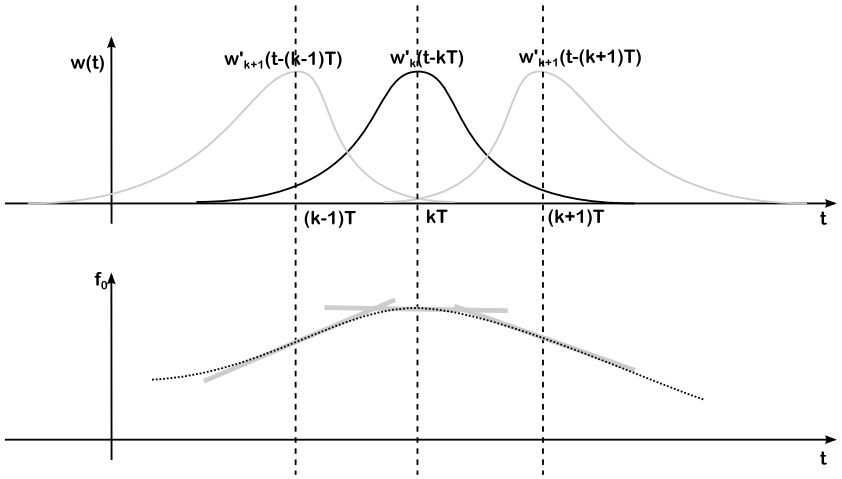


FIGURE 7.2: Schematic of three consecutive windows with different chirp rates.

have been proposed to reconstruct the audio, as presented in [61] or in [62] as improvements to the simple Overlap-Add. If the STFT parameters are chosen appropriately, satisfying the constant-overlap-add constraint, perfect reconstruction can be achieved. This is the case when using 50% window overlapping, and a Hanning window to calculate the transform, in which the classical Overlap-Add technique allows for a simple and effective way of reconstructing the signal.

The FChT, can also be used as a low-level data representation for source separation algorithms, which is especially suitable in the case of voiced speech or harmonic signals. As the results of these algorithms may be expressed in terms of the components of the FChT, it would be a useful method to reconstruct a signal based on a Short Time FChT. Equivalently to what happens when using the STFT, in the sense that actually a Modified STFT is inverted, the reconstruction may not correspond to the a real STFChT but to a Modified Short Time FChT which may not take consistent values, as this representation is also redundant as the STFT.

The following example illustrates the source separation process. A signal $x(n) = x_1(n) + x_2(n)$ is considered, consisting of two linear harmonic chirps $x_1(n)$ y $x_2(n)$, with different chirp rates. Each chirp signal has four harmonics. Signals $x_1(n)$, $x_2(n)$ and $x(n)$ are shown in Figure 7.3. The FChT computation process for the signal $x(n)$ is depicted in Figure 7.4. The chirp rate value of the transform was selected to match the chirp rate of the signal $x_1(n)$. Figure 7.4a shows the warping process. The spectrum of the original and the warped signals are presented in Figure 7.4b. Since the fundamental frequency changes rapidly within the analysis window, the spectral peaks are wide and overlap, as can be seen in the spectrum

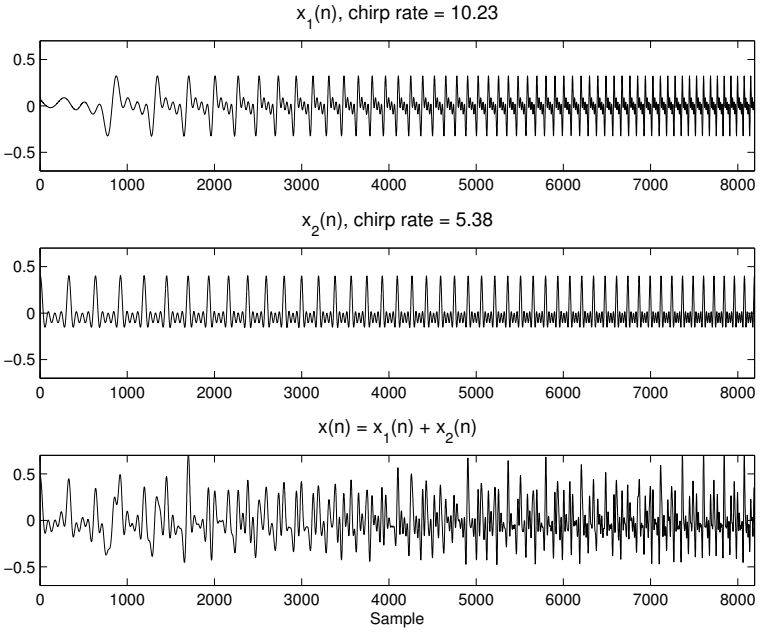
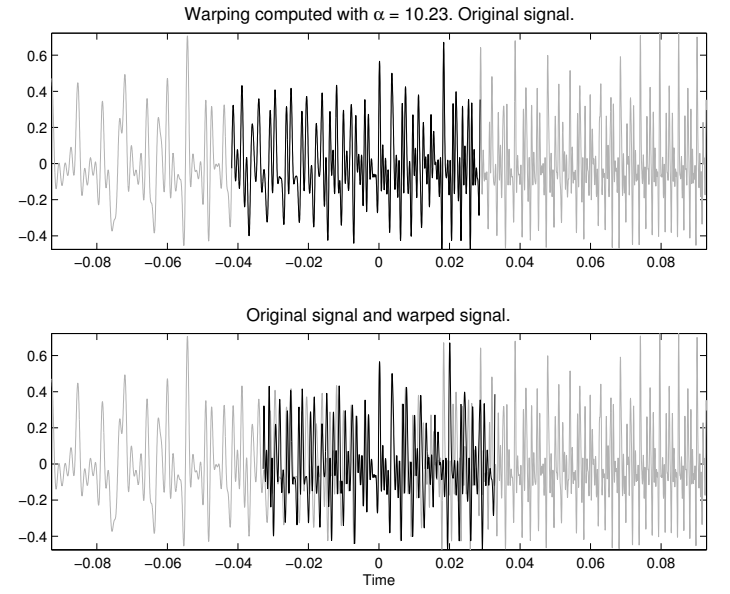


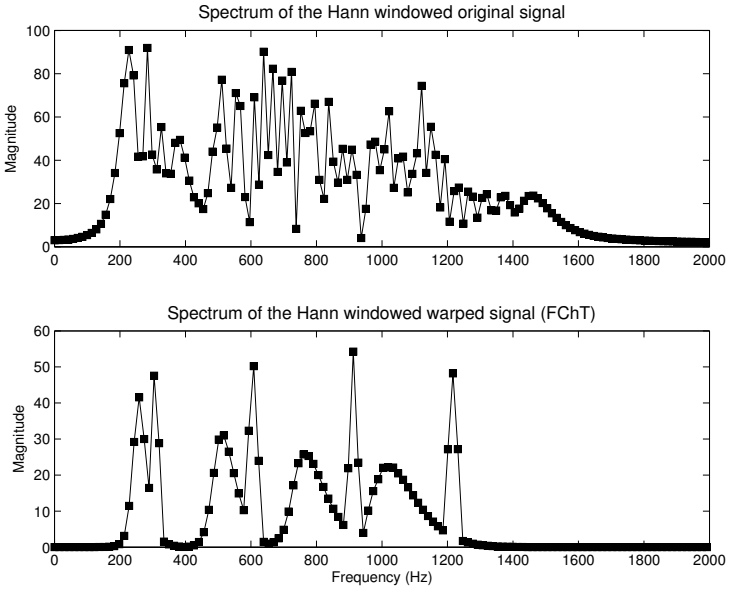
FIGURE 7.3: Example of the use of the FChT in source separation. The signal $x(n)$ consists of the sum of two harmonic chirps $x_1(n)$ and $x_2(n)$ with a strongly different chirp rate α of 10.23 and 5.38 respectively.

of the original signal. The interference produced by the overlapping peaks makes it difficult to properly discern which components correspond to each chirp signal. Conversely, the harmonic chirp $x_1(n)$ becomes a stationary harmonic signal after the warping process, so its components have a minimum bandwidth enabling a better discrimination of the spectral peaks.

Once the FChT is tuned to a certain sound source, namely the target source, narrow-band spectral peaks are obtained. It is possible to isolate these peaks from the spectrum of the audio mix by using a simple spectral mask. Then, the target source can be resynthesized by means of the inverse FChT. This is illustrated in Figure 7.5. The masking is carried out by a narrow-band filter-bank, as depicted in Figure 7.5a. In this example the bandwidth of each filter is 3 bins. This is because a Hann window is applied in the FChT analysis, and a Hann-windowed sinusoid has a spectral band-width of 3 bins (provided that an exact number of cycles fit into the analysis window). The result of the separation process is shown in Figure 7.5b. The target signal $x_1(n)$ is synthesized by the inverse FChT of the filtered spectrum. By subtracting it from the original signal a residual is obtained, that in this case is the signal $x_2(n)$. Some remarks with regards to the results are in order. Firstly, the synthesized signal obtained is amplitude modulated by the time-warped Hann window, as described in Section 4.1.1. Therefore, in order to

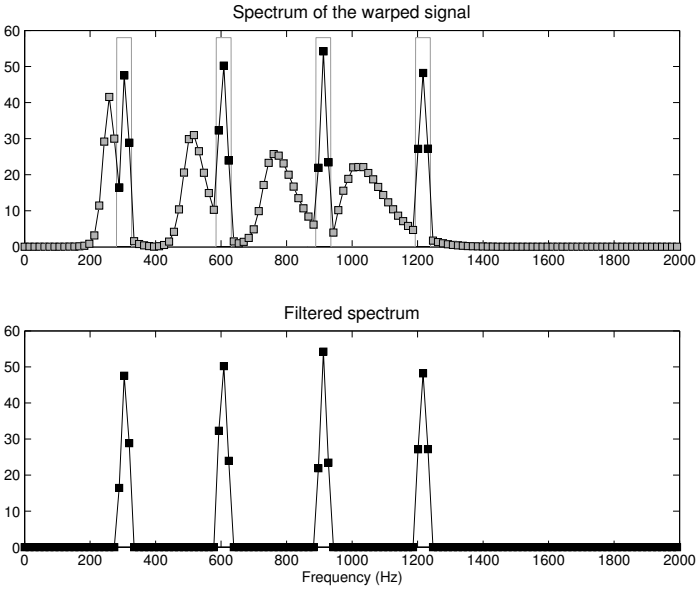


(A)

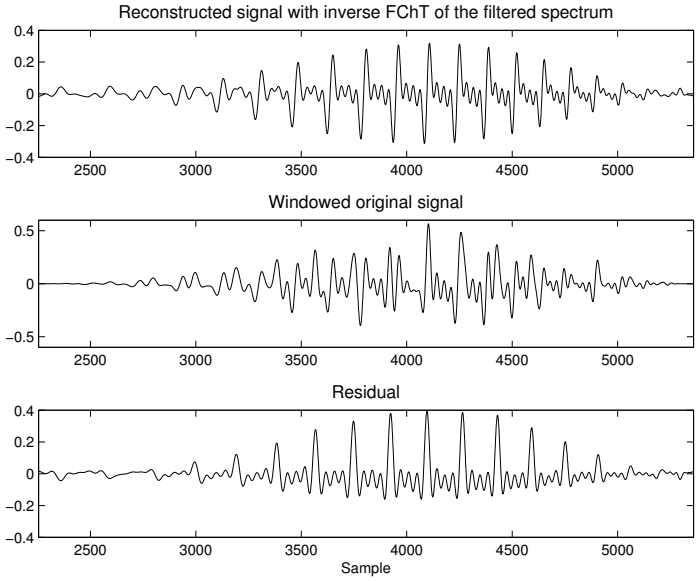


(B)

FIGURE 7.4: Comparison between the spectrum of the original and warped signals. Waveform of the original signal and the signal deformed according to the pitch change of the signal. In the spectrum of the original signal, the peaks are spread and overlap, while in the warped signal they exhibit minimum bandwidth enabling to distinguish their energy from the rest of the mix.



(A)



(B)

FIGURE 7.5: Separation of signals $x_1(n)$ and $x_2(n)$ by filtering the FChT and applying the inverse transform. (a) Spectral peak selection of $x_1(n)$ by a set of harmonic filters, 3 bins wideeach. (b) Inverse transform of the filtered signal to recover $x_1(n)$. The residual is obtained by subtracting this synthesized signal from the mix.

obtain $x_2(n)$, the original signal has to be smoothed using the same time-warped Hann-window. However, in practice, the target signal is reconstructed by overlap-add and the amplitude modulation effect produced by the windowing process can be compensated. Secondly, although the reconstructed signals should be identical to the windowed original signals, some differences can be observed, mainly at the edges of the window. This differences arise from, on one hand, the interference of the rest of the mixture on the filtered spectrum, and on the other hand, spectral components of the target source that lie outside the band-width of the filters.

7.3 Experimental Results

In order to evaluate the separation power of the method, the output of a system based in the FChT was submitted to the Signal Separation Evaluation Campaign[1] (SiSEC). This is a contest to evaluate signal separation techniques on different tasks, some of them regarding speech and music. The task "Professionally produced music recordings", consists in isolating the leading voice from a given piece of music. The participants are asked to submit the audio of the isolated sources which are unknown to them. The organizers of the campaign evaluate the quality of the separation based on the submitted audio.

The algorithm based on the FChT to separate harmonic sources from a musical mix was used to separate the voiced content of the audio. Although most of the energy of the leading voice is present in the voiced sounds, for the time intervals when only unvoiced sounds were present an ad-hoc bandpass filtering was implemented

The evaluation of the separation of two recordings are published in the SiSEC webpage [2] with very good results. In one of the recordings, the performance was a SIR for the left and right channels (L,R) of (23 dB, 25.2 dB), a SAR of (9.7 dB, 9.7 dB), and in the second one a SIR of 17.7 dB and a SAR of 5.2 dB, which are results among the best results obtained in the campaign. Figure 7.6 shows the spectrogram of the original left channel of the first recording, and the spectrogram of the extracted leading voice.

This reinforces the idea that the FChT can be successfully used as a tool for signal source separation of harmonic sources.

[1] http://sisec2010.wiki.irisa.fr/tiki-index.php
[2] http://www.irisa.fr/metiss/SiSEC08/SiSEC_professional/

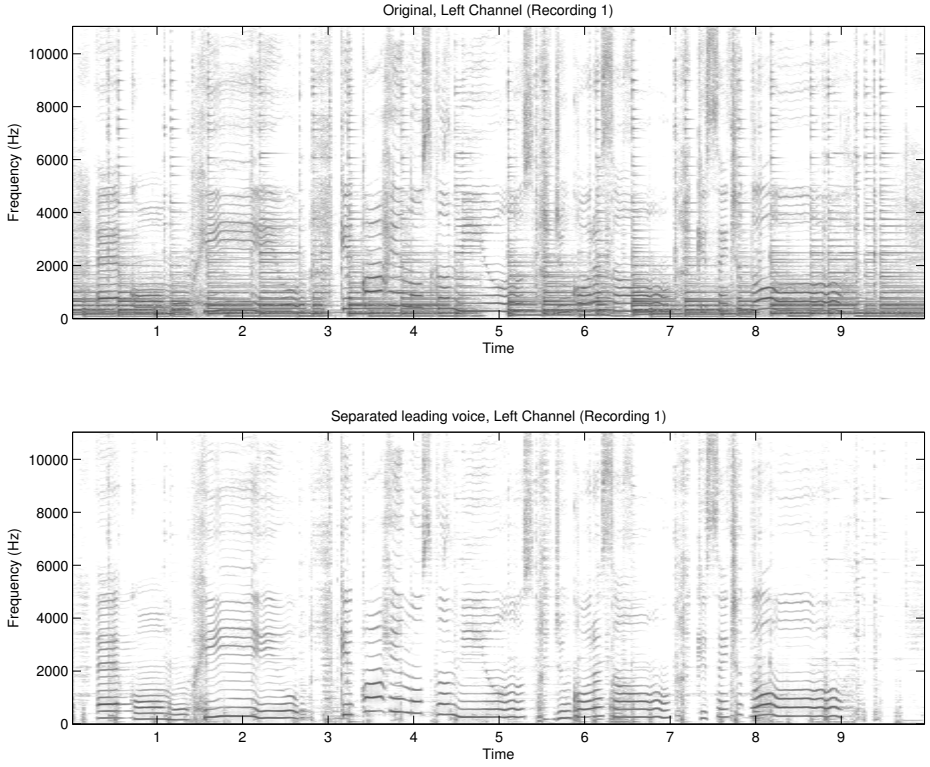


FIGURE 7.6: Top: original spectrogram of the mix of the first recording of the SiSEC "Professionally produced music recordings" task. Bottom: the extracted leader voice spectrogram.

7.4 Conclusions and Future Work

The representation of a harmonic source with the correct chirp rate α gives a sparse representation that concentrates the energy of the signal in few coefficients. This property makes Fan-Chirp Transform to be a good tool to isolate a harmonic source from a mix with a simple comb filter. While some nonlinear operations are performed in the calculation of the FChT, it is possible to compute the inverse process which combined in an Overlap-Add fashion, produces excellent separation results. The obtained separated source is coherent (in phase) with that on the mix, this makes possible to calculate the residual of the separation by a simple subtraction of the isolated source from the mix.

In some voiced sounds, especially in speech or a singing voice, there is also an unvoiced component that has a broad band signal that is produced by a turbulent flow of air. This is the typical component that appears when whispering. The FChT works well modeling the harmonic part, which can be separated properly

from the mix, but a"whispered" sound of the extracted source stays in the residual. The spectral envelope of the separated source could be used to estimate the unvoiced residual part as both the voiced and unvoiced components have a similar spectral envelope determined by the same vocal tract resonance.

The proposed separation algorithm does not take any other assumptions of the signal but its harmonicity. It would be interesting to use a model of the spectral envelope such as Gaussian Mixture Models as was described and used in Section 6. The separation results could be improved especially in the low frequency range (< 400Hz) as the absolute frequency change for any chirp rates is small which makes hard to discriminate the actual amount of energy that comes from a source when other sources with similar fundamental frequency are present.

Chapter 8

FChT Source Separation for a Query by Humming Application

In this chapter we focus on the usage of the signal source separation based on the FChT to isolate the main melody of a song to be used as the input to populate an already existing QBH system. In this way, the FChT is used as a low level representation by the application presented in Chapter 7 as a part of a bigger end-user application.

The contents of this Chapter are based on the article [24] published in collaboration with Martín Rocamora and Álvaro Pardo. The description reproduces some the article passages, as well as includes some modifications or additions in order to contribute to the structure of this document.

8.1 Introduction

The constant increase in computer storage and processing capabilities has made possible to collect vast amounts of information, most of which is available online. Today, people interact with this information using various devices, such as desktop computers, mobile phones or PDAs, posing new challenges at the interface between human and machine. Yet, the most common case of information access still involves typing a query to a search engine. There is a need for new human-machine interaction modalities that exploit multiple communication channels to make our systems more usable.

Among the information available there are huge music collections, containing not only audio recordings, but also video clips and other music-related data such as text (e.g. tags, scores, lyrics) and images (e.g. album covers, photos, scanned sheet music). A query for music search is usually formulated in textual form, by including information on composer, performer, music genre, song title or lyrics. However, other modalities to access music collections can also be considered that allow more intuitive queries. For instance, to provide a musical excerpt as an example and obtain all the pieces that are similar in some sense, namely query-by-example,[1] or to retrieve a musical piece by singing or humming a few notes of its melody, which is called query-by-humming (QBH). This offers an interesting interaction possibility, in particular for small size devices such as portable audio players, and requires no music theoretical knowledge from the user. Additionally, it can be combined with traditional metadata-based search and visual user interfaces to offer multimodal input and output, in the form of visual and auditory information.

Dealing with multimodal music information requires the development of methods for automatically establishing semantic relationships between different music representations and formats, for example, sheet music to audio synchronization or lyrics to audio alignment [63]. Much research in audio signal processing over the last years has been devoted to music information retrieval [64, 65], i.e. the extraction of musically meaningful content information from the automatic analysis of an audio recording. This involves diverse music related problems and applications, from computer aided musicology [34], to automatic music transcription [66] and recommendation [67]. Many research efforts have been devoted to dealing with the singing voice, tackling problems such as singing voice separation [68] and melody transcription [69]. The incorporation of these techniques into multimodal interaction systems can lead to novel and more engaging music learning, searching and gaming applications.

Even though the problem of building a QBH system has received a lot of attention from the research community for more than a decade [70], the automatic generation of the melody database against which the queries are matched is still an open issue. In most of the proposed systems the database consists of music in symbolic notation, e.g. MIDI files. This is due to the lack of sufficiently robust automatic methods to extract the melody directly from a music recording. Although there is a great amount of MIDI files online, music is mainly recorded and distributed as audio files. Hence, the scope of this approach is limited because of the need

[1] Audio fingerprinting techniques are used in this case, being Shazam (http://www.shazam.com/) probably one of the best known commercial services of this kind.

of manually transcribing (i.e. audio to MIDI) every new song of the database. A way to circumvent this problem is to build a database of queries provided by the users themselves and to match new queries against the previously recorded ones [71]. This approach drastically simplifies the problem and is applied in music search services such as SoundHound.[2] However, the process is not automatic but relies on user contributions. Besides, a new song can not be found until some user records it for the first time. In order to extend QBH systems to large scale it is necessary to develop a full automatic process to build the database. There are only a few proposals of a system of this kind [72–75] and results indicate there is still a lot of room for improvement to reach the performance of the traditional systems based on symbolic databases.

In this section we describe a method for automatically building the database of a QBH, in which the singing voice melody is extracted from a polyphonic music recording. The automatic construction of the database is based on the melody extraction from a FChT based analysis. A prototype is built as a proof of concept of the proposed method and the performance of a previously developed QBH system [76] is evaluated when using a database of MIDI files and when using melodies extracted automatically from the original recorded songs.

Next section briefly describes the QBH system used in the experiments. The method for extracting the singing voice melody from polyphonic music recordings is presented in section 8.3. In Section 8.4 the experiments carried out for assessing the performance of the QBH on the automatically obtained database are described and some results are reported.

8.2 Query-by-humming system

The existing QBH systems can be divided, from its representation and matching technique, basically into two approaches. The most typical solution is based on a note by note comparison [77, 78]. The query voice signal is transcribed into a sequence of notes and the best occurrences of this pattern are identified in a database of tunes (typically MIDI files).

The melody matching problem poses some challenges to be considered. A melody can be identified in spite of being performed at different pitch and at different tempo. Additionally, sporadic pitch and duration errors or expressive features

[2] http://www.soundhound.com/

modify the melodic line but still allow the melody to be recognized. In the matching step, pitch and tempo invariance are typically taken into account by coding the melodies into pitch and duration contours. By means of flexible similarity rules it is possibly to achieve some tolerance to singing mistakes and automatic transcription errors. Automatic transcription of the query inevitably introduce errors that tend to deteriorate matching performance. For this reason, another usual approach avoids the automatic transcription, comparing melodies as fundamental frequency (F0) time series [79, 80]. Unfortunately, this involves working with long sequences, very long compared to note sequences, and therefore it implies high computational burden. Moreover, in many proposals the user is required to sing a previously defined melody fragment [79, 80] in order that the query exactly matches an element of the database. This is because of the difficulty of searching subsequences into sequences providing pitch and tempo invariance.

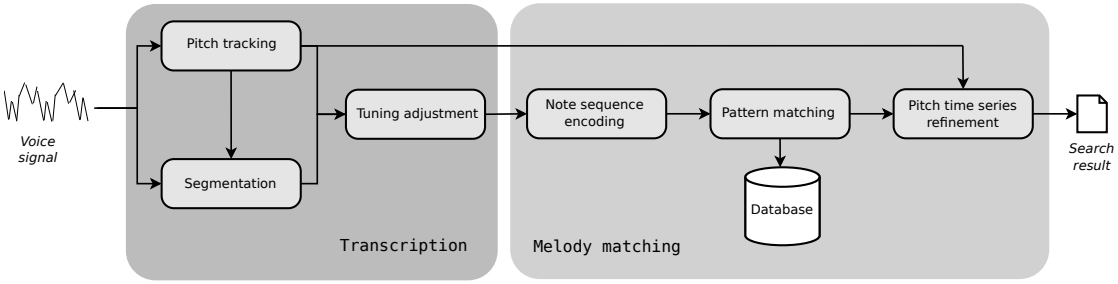


FIGURE 8.1: Block diagram of the QBH system. The input is a monophonic singing voice. The two main stages are the transcription of the query and the match of the melody pattern against the elements of the database. The output is a ranked list of songs.

In [76] Rocamora et al. presented a way of combining both approaches, exploiting the advantages of each of them. Firstly, the system selects a reduced group of candidates from the database using note by note matching. Then, the selection is refined using fundamental frequency time series comparison. Finally, a list of musical pieces is retrieved in a similarity order. Figure 8.1 shows a block diagram of the system.

The system architecture is divided in two main stages. The first one is the transcription of the query into a sequence of notes. In the second stage, the notes of the query are matched to the melodies of the database. Finding good occurrences of the codified query in the database is basically an approximate string matching problem. For this task, Dynamic Programming is used to compute an edit distance that combines duration and pitch information [81]. In this combination, pitch values are considered more important because duration information is less

discriminative and not so reliable. More details on the evaluation of the similarity between the sequences can be checked in [76].

The edit distance, $d_{i,j}$, is computed recursively as the minimum of the set of values shown in Equation 8.1.

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + 1, & & \text{(insertion)} \\ d_{i,j-1} + 1, & & \text{(deletion)} \\ d_{i-1,j-1} + 1, & & \text{(note substitution)} \\ d_{i-1,j-1} - 1, & |\overline{a}_i - \overline{a}'_j| < 2 \text{ and} & \text{(coincidence)} \\ & |q_i - q'_j| < 2 & \\ d_{i-1,j-1} & |\overline{a}_i - \overline{a}'_j| < 2 & \text{(duration substitution)} \end{cases} \tag{8.1}$$

The last two values of the set are only considered if the corresponding conditions are met, where $\overline{a}$ and $\overline{a}'$ refer to the pitch interval of the query and the database element respectively, whereas q and q' correspond to their quantized relative duration. Finally, a similarity score is computed normalizing the edit distance to take values between 0 and 100,

$$\text{score} = 100 \, \frac{(m-1) - d_{m,m}}{2(m-1)} \tag{8.2}$$

where m denotes the number of notes in the query, and $d_{m,m}$ is the final value of the edit distance between the two sequences.

As a result of the notes sequence matching, fragments similar to the query pattern are identified in the melodies of the database. Then F0 time series of this fragments are built from the matching MIDI notes, and are compared to the F0 contour of the query by means of Local Dynamic Time Warping (LDTW). The sequences are time warped to the same duration and pitch transposed to the same tunning.

In this way, LDTW is applied to a small group of candidates (10 for the reported results), which is computationally efficient, and without imposing constrains to the query, since coincident fragments are identified automatically in the notes matching stage. Figure 8.2 shows an example of the comparison of note sequences and F0 time series between the query and an element of the database.

Even though the search is efficient, given the two-stage matching approach, the notes matching performs an exhaustive scan of the database that can become prohibitive in a large scale scenario. This may be tackled with hashing techniques as in [82].

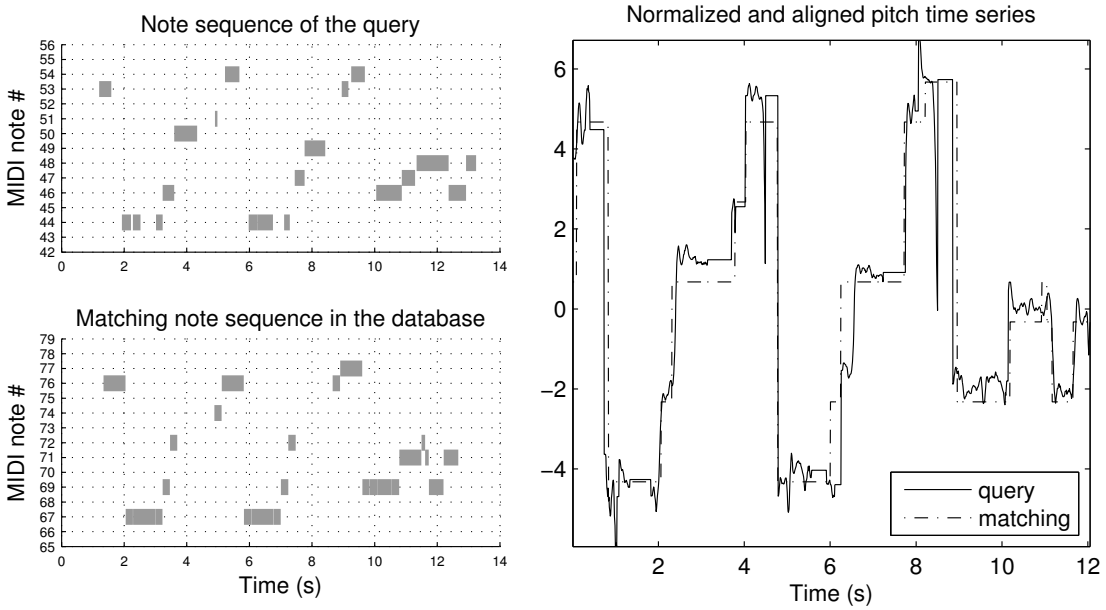


FIGURE 8.2: The piano-roll representations to the left show the transcription of the query at the top, and of an occurrence in the database at the bottom. The plot to the right depicts the corresponding F0 time series normalized and aligned by the system.

8.3 Singing voice melody extraction from polyphonic music

For building the database we focus on extracting the singing voice melody from the original polyphonic music recordings, based on the hypothesis that the melody of the leading voice is the most memorable and distinctive tune of the song and would most probably be used as a query[3]. In a first step, the pitch tracking algorithm described in 5.2 is used to find the fundamental frequency of the most prominent melody. The audio that corresponds to that source is separated form the polyphonic mix using the method described in Section 7.2. After that, audio features are computed for each of the extracted sounds and they are classified as being singing voice or not, as described in [84]. The sounds classified as vocal are mixed in a mono channel and the transcription method used in the QBH system for transcribing the query is applied to obtain a sequence of notes and a F0 contour. This information is indexed as an element of the database. The process is depicted in Figure 8.3 and described in the following sections.

[3]According to [83], there is experimental evidence that indicates that the memory representation for lyrics seems to be tied into the memory representation for melody, providing multiple redundant constraints to assist the recall of a passage.

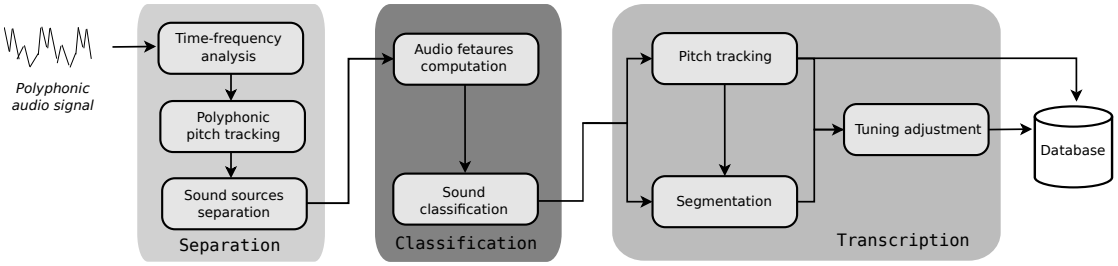


FIGURE 8.3: Block diagram of the process for automatically building the database. The system involves three main steps: Separation, Classification and Transcription. The subtasks of each step are also indicated. Note that the same monophonic transcription block used for processing queries in the QBH system is applied.

8.3.1 Harmonic sounds separation

The time–frequency analysis is based on the Fan Chirp Transform which is especially well suited for singing voice analysis since most of its sounds have a harmonic structure and their frequency modulation can be approximated as linear within short time intervals.

In addition, the F0gram described in Section 3.3.1 is used to describe the pitch salience, which reveals the evolution of pitch contours in the signal, as depicted in Figures 8.4 and 8.6. Given the FChT of a frame $X(f, \alpha)$, salience (or prominence) of fundamental frequency f_0 is obtained by summing the log-spectrum at the positions of the corresponding harmonics,

$$\rho(f_0, \alpha) = \frac{1}{n_H} \sum_{i=1}^{n_H} \log |X(if_0, \alpha)|, \tag{8.3}$$

where n_H is the number of harmonics considered. Polyphonic pitch tracking is carried out by means of the technique described in 5.2.3, which is based on unsupervised clustering of F0gram peaks. Finally, each of the identified pitch contours are separated from the sound mixture. To do this, the FChT spectrum is band–pass filtered at the location of the harmonics of the f_0 value, and the inverse FChT is performed to obtain the waveform of the separated sound.

8.3.2 Singing voice classification

The extracted sounds are then classified as proposed in [84], based on classical spectral timbre Mel-frequency Cepstral Coefficients (MFCC) features and some

features selected to capture characteristics of typical singing voice pitch contours. The implementation of MFCC is based on [85]. Temporal integration is done by computing median and standard deviation of the frame-based coefficients within the whole pitch contour. First order derivatives of the coefficients are also included to capture temporal information, obtaining a total of 50 audio features. In a musical piece, pitch variations are used by a singer to convey different expressive intentions and to stand out from the accompaniment. Most typical expressive features are *vibrato*, a periodic pitch modulation, and *glissando*, a slide between two pitches [86]. Thus, low frequency modulations of a pitch contour are considered as an indication of singing voice. Nevertheless, since other musical instruments can produce such modulations, this feature is combined with other sources of information.

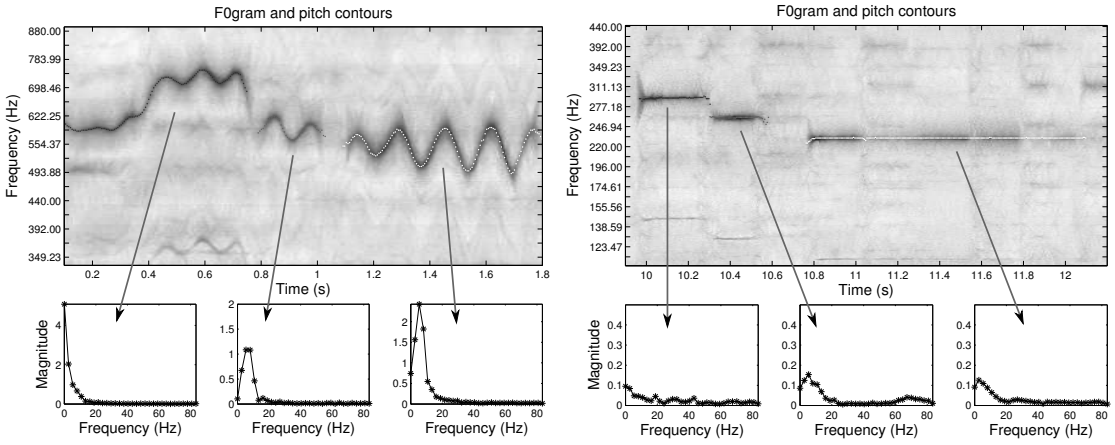


FIGURE 8.4: Vocal notes with vibrato and low frequency modulation (*left*) and saxophone notes without pitch fluctuations (*right*) for two audio files from the MIREX [22] melody extraction test set. Summary spectrum $\tilde{c}[k]$ is depicted at the bottom for each contour.

In order to describe the pitch variations, the contour is regarded as a time dependent signal $f_0[n]$ and a spectral analysis is applied using the DCT. Examples of the behavior of the spectral coefficients, $\tilde{c}[k]$, are given in Figure 8.4. The two following features are derived from this spectrum,

$$\text{LFP} = \sum_{k=1}^{k_L} \tilde{c}[k], \quad \text{PR} = \frac{\text{LFP}}{\sum_{k_L+1}^{N} \tilde{c}[k]}. \tag{8.4}$$

The low frequency power (LFP) is computed as the sum of absolute values up to 20 Hz ($k = k_L$) and reveals low frequency pitch modulations. The low to high frequency power ratio (PR) additionally exploits the fact that well-behaved pitch contours do not exhibit prominent components in the high frequency range.

Besides, two additional pitch related features are computed. One of them is simply the extent of pitch variation,

$$\Delta f_0 = \max_n\{f_0[n]\} - \min_n\{f_0[n]\}. \tag{8.5}$$

The other is the mean value of pitch salience in the contour,

$$\Gamma_{f_0} = \underset{n}{\text{mean}}\{\rho(f_0[n])\}. \tag{8.6}$$

This gives an indication of the prominence of the sound source, but it also includes some additional information. As noted in Section 3.3.1, pitch salience computation favors harmonic sounds with high number of harmonics, such as the singing voice. Additionally, as done in [50], a *pitch preference* weighting function is introduced that highlights most probable values for a singing voice in the f_0 selected range.

The training database is based on more than 2000 audio files, comprising singing voice on one hand and typical musical instruments found in popular music on the other. For building the database the sounds separation front-end is applied (i.e. the FChT analysis followed by pitch tracking and sound source extraction) and the audio features are computed for each extracted sound. In this way, a database of 13598 sound elements is obtained, where vocal and non-vocal classes are exactly balanced. Histograms and box-plots are presented in Figure 8.5 for the pitch related features on the training patterns. Although these features should be combined with other sources of information, they are informative about the class of the sound. An SVM classifier with a Gaussian RBF Kernel was selected for the classification experiments, using the Weka software [87]. Optimal values for the γ kernel parameter and the penalty factor C were selected by grid-search [88].

8.3.3 Singing voice melody transcription

Finally, the sounds classified as singing voice are mixed in a single mono audio channel and the same transcription procedure used for processing the queries is applied. This yields the singing voice melody out from the polyphonic music recording, as a sequence of notes and as a pitch contour. Figure 8.6 shows the whole process for a short audio excerpt of the song *For no one* by The Beatles, which belongs to the automatically built database of the QBH system.

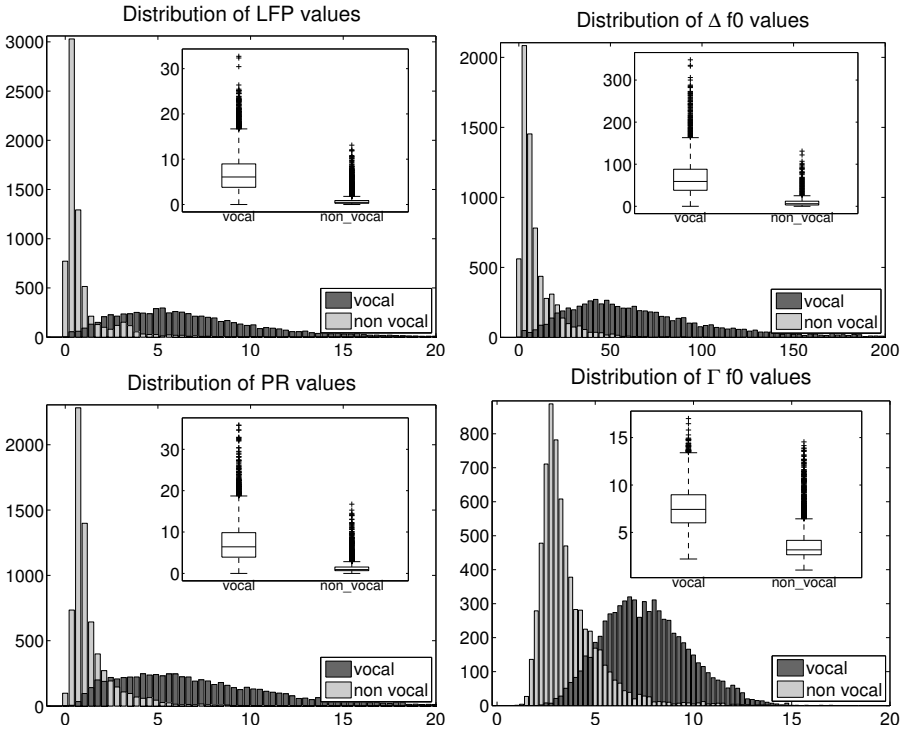


FIGURE 8.5: Histograms and box-plots of the pitch related feature values on the training database for the vocal and non-vocal classes.

8.4 Experiments and results

8.4.1 Experimental setup

The experiment is designed to evaluate the validity of extending an existing MIDI files database by using the proposed automatic method. To do that, two different datasets are used. The first one is a collection of 208 MIDI files corresponding to almost all the songs recorded by The Beatles (excluding duplicates and instrumentals) gathered from the Internet.[4] This music was selected because it is widely known making it easy to get volunteers for queries, it has generally a clear and distinctive singing voice melody, and is readily available both in audio and MIDI.

The melody of a song is assumed to be the one performed by the leading singing voice, which is usually a single MIDI channel labeled as *leading voice* or *melody*. This channel is manually extracted and indexed as an element of the database. To build the second database, 12 songs are selected out of this collection (which are listed in Table 8.1), and their melody is automatically extracted from a mono mix of the audio recording. The selection comprises different music styles and

[4]From websites such as *The Beatles MIDI and video heaven*, http://beatles.zde.cz/.

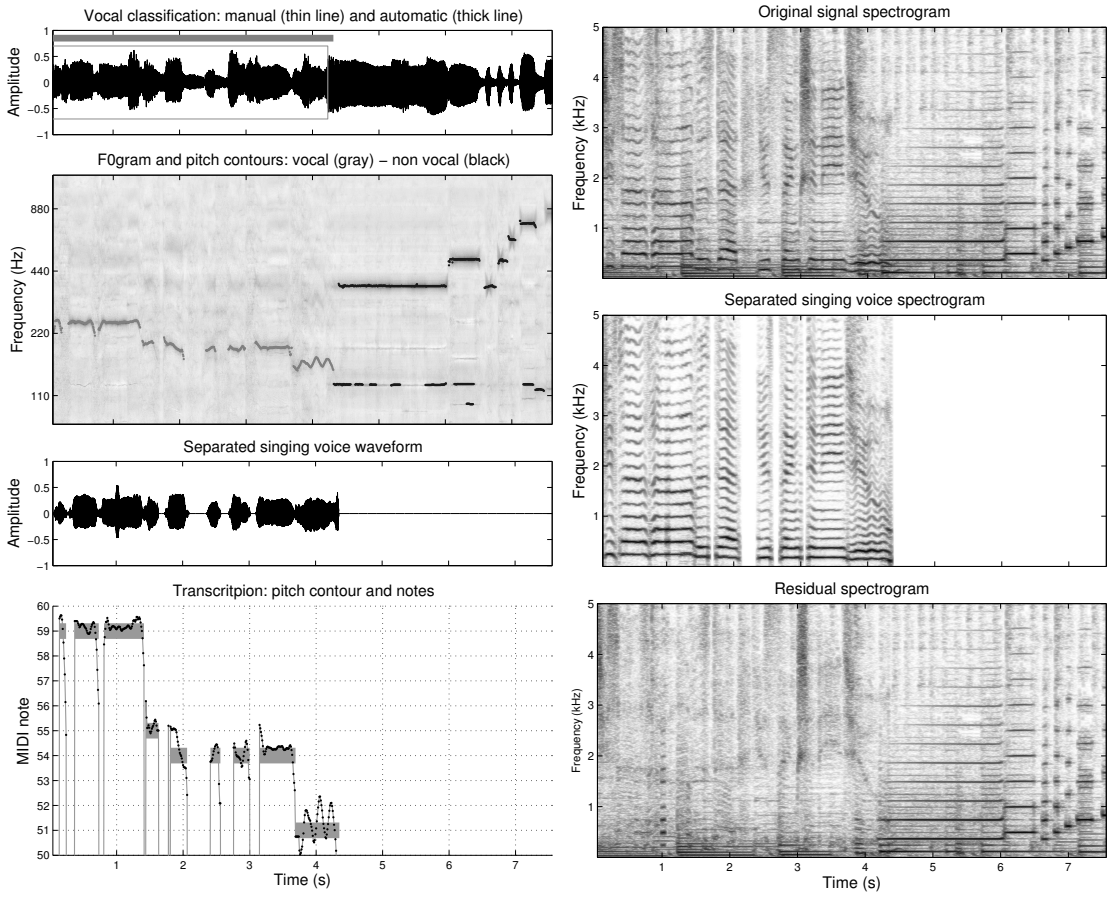


FIGURE 8.6: Example of the automatic process for building the database using a fragment of the song *For no one* by The Beatles. A singing voice in the beginning is followed by a French horn solo. There is a soft accompaniment of bass and tambourine. On the left, from top to bottom: the waveform of the recording (with manual and automatic vocal labeling), the F0gram showing both vocal and other sources pitch contours (automatically labeled), the extracted singing voice waveform, and the transcription to notes and F0 contour of the extracted singing voice. On the right, the corresponding spectrograms of the original audio mix, the extracted singing voice and the residual (the extracted singing voice subtracted from the mix).

instrumentations (e.g. rock & roll, ballads, drums, bowed strings), but does not include too dense polyphonies in order that the singing main melody could be identified with no difficulty by a listener. In this case the database is modified by replacing the manually created MIDI files by the automatically extracted melodies (notes sequence and pitch contour) for the aforementioned songs.

A set of 106 sung queries corresponding to the selected songs was recorded by 10 not trained singers (6 male and 4 female), using standard desktop computer hardware. The participants were asked to sing the melody as they remembered it, with no restrictions on singing only a vocal part. They were free to sing with

TABLE 8.1: Experimental setup. List of the 12 selected songs whose melody is automatically obtained.

Song title
Blackbird
Do you want to know a secret
For no one
Girl
Hey Jude
I call your name
I've just seen a face
Michelle
Rocky raccoon
The fool on the hill
When I'm sixty four
Yesterday

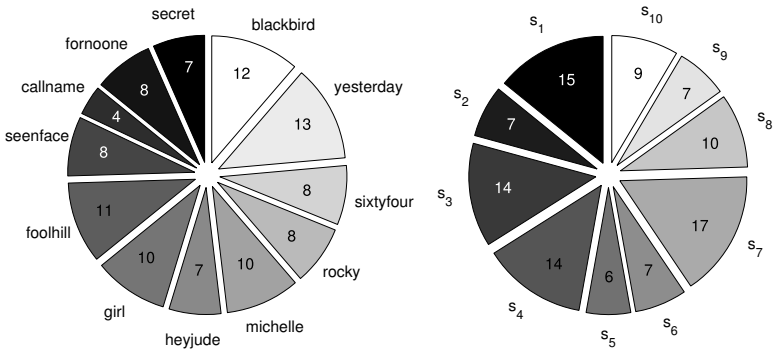


FIGURE 8.7: The left-side chart shows the distribution of queries among the selected songs. The distribution of queries among the 10 singers s_i is depicted in the right-side chart. Note that the database is well balanced in both aspects.

lyrics, hum (with syllables such as 'ta' or 'la'), or a combination of both. The mean number of notes in a query is 28, and the distribution of queries among the songs and singers is shown in Figure 8.7. The whole set of queries is available online, along with the mono mix and the automatic transcription of the selected songs.[5] Although including queries that do not correspond to the set of replaced songs may potentially give more insight of the QBH system, it makes the analysis of the database extension more troublesome and therefore will not be reported.

[5]Available from http://iie.fing.edu.uy/investigacion/grupos/gpa/QBH/.

8.4.2 Query by humming evaluation

In order to evaluate the performance of the QBH system two standard measures are adopted: mean reciprocal rank (MRR) and top-X hit rates. Let r_i be the rank of the correct song in the retrieved list for the i-th query. Top-X hit rates are the proportion of queries for which $r_i \leq X$. Considering a set of N queries, the MRR is computed as,

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}. \tag{8.7}$$

Two different alternatives are considered for the audio based database. Recall that the system performs a final refinement by the direct comparison of F0 time series devised to improve matching performance. This refinement avoids errors introduced in the automatic transcription of the query. When a database of MIDI files is used, F0 time series of the matching candidates are built from the pitch of MIDI notes. In the case of the audio based database, errors are also introduced in the transcription of the singing voice melody extracted from the recording (see section 8.3.3). Therefore, it is preferable to perform the refinement using F0 time series computed from the extracted singing voice, rather than building it from the transcribed notes. This is confirmed by the results shown in Table 8.2, where the two different LDTW refinements are considered. Since the refinement is done over the 10 best matching candidates, top-10 hit rates remain unchanged.

TABLE 8.2: QBH evaluation results for MIDI and audio based databases. For the latter, the query is aligned to two different F0 time series of the matching candidate: the pitch of the transcribed notes (audio 1) and the extracted F0 contour (audio 2). Recall that the number of queries is 106 and the total number of songs (different classes) is 208.

	MRR	Top-X hit rate (%)		
		1	5	10
MIDI	0.89	88.68	89.62	91.51
audio 1	0.75	69.81	79.25	84.91
audio 2	0.76	71.70	81.13	84.91

As a way of further comparing both types of databases, an analysis is conducted considering the notes matching score assigned to the retrieved items (see Equation 8.2). For each query, the score of the correct song is plotted against the highest score of the wrongly retrieved elements, as shown in Figure 8.8. This is intended to study the ability of the score to discriminate between correct and wrong retrieves. A top-1 hit result implies a correct song score higher than all the others. Thus,

ideally all the query points would be located in the right-bottom triangle of the graph. For the MIDI database the vast majority of elements lie in that region, particularly for higher correct song scores. While not so markedly, the behaviour is similar for the audio based database. In the light of the above, a threshold on the score value can be useful as way of assuring confidence on the results. The thresholding determines the typical binary class scenario, resulting in True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) regions, as depicted in Figure 8.8. This allows the comparison of the methods using a ROC curve, also shown in the figure. Although the MIDI database gives better results, the performance of the audio based databased is promising. As for illustrative purposes only, operating points are depicted as filled markers in the ROC (the farthest point to the diagonal), and their corresponding thresholds are plotted as vertical lines.

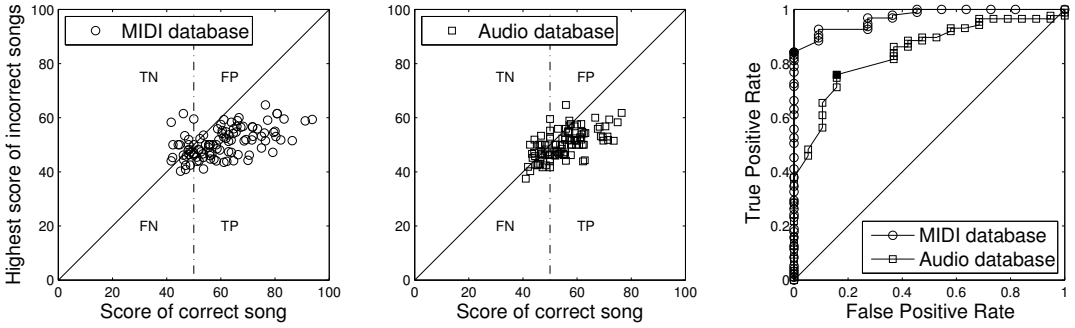


FIGURE 8.8: Analysis of the information given by the score assigned to the retrieved items for the MIDI and the audio databases. The plots to the left and center show the score of the correct song against the highest score of the wrongly retrieved elements for each query. The plot on the right shows a ROC curve for each database obtained by using different thresholds on the score value.

8.5 Discussion and conclusions

A multimodal interface for music retrieval was considered, in which the user sings or hums a few notes of a melody as a query. The main drawback of these QBH systems is their difficult scalability, since manual annotation is required to build the database. A method based in the FChT was proposed to tackle this problem making it possible to extend an existing database automatically from audio recordings. A prototype of a complete system was developed in order to test the validity of the proposal. The experiments conducted show that the matching performance achieved is considerably high, obtaining 85% of the correct item in the

top-10. Besides, the information provided by the scores assigned to the matching items can be exploited to determine the confidence in the retrieval.

As expected, the automatic singing melody extraction from audio recordings is not as accurate as the manual transcription, and this in turn decreases the performance of the QBH system. Nevertheless, even though the top-1 hit rate is significantly affected, the difference becomes less important for the top-10 and it is still above the reported rate for humans attempting to identify queries by ear (66%) [89]. Moreover, the evaluation of the audio based system yields an MRR of 0.76 for a database of 208 songs and 106 queries. Although a fair comparison between different experiments is not possible, the performance is encouraging given the best results for similar setups reported in other works (e.g. an MRR of 0.58 for a database of 427 songs and 159 queries [74], and an MRR of 0.56 for a database of 481 songs and 118 queries [75]).

While there is still room for improvement, the FChT proved to be an adequate tool to extract a melody that can be used in applications such as a QBH system.

Chapter 9

Discussion and Future Work

Different applications based on the FChT were analyzed, showing the potential of the transform to be used as a low-level representation. A simple and flexible implementation of the FChT has been made freely available as an open source Matlab package, and as a Sonic Visualizer Plug-In.

The Time–Frequency representation of musical signals was studied based on an independent formulation of the Fan Chirp Transform with some differences in the use of the analysis window with respect to [13]. This optimizes the representation properties in the warped domain, but produces a slight time shift that can be easily corrected. The properties of the windowing and time warping were analyzed and the limitations of the method were described. Different extensions were considered to improve the representation of harmonic signals in the F0-gram Time–Frequency map of harmonic signals. The FChT was combined with the constant Q transform which gives a slightly better representation for higher harmonics. Also, the utilization of nonlinear warpings was studied, showing the benefits and limitations of their use. The analysis of simultaneous sound sources was addressed for which a post-processing of the F0-gram is proposed to remove spurious multiples and sub-multiples improving the representation multiple harmonic sounds.

One of the studied applications is the use of the F0-gram as a tool to allow the musicological analysis of the pitch evolution in polyphonic music which has been illustrated by the study of two musical pieces by Luis Jure. An open source freely available plug-in and a Matlab software have been made available to be used for such purposes. The representation is also used to extract the pitch of the main melody of a musical piece. The FChT gives local candidates of the melody contour. The tracking of these candidates improves the results, eliminating

spurious candidates due to other sources present in the mixtures. The slope of the pitch of a candidate was used to define a similarity measure between pairs of time adjacent candidates. This similarity measure enables the possibility to use some classical clustering techniques to group candidates from the same source, and thus can be used to track the main melody. The results were evaluated using the ground truth available for the MIREX Main Melody Extraction databases MIREX04 and MIREX05 giving promising results.

The automatic detection of the main melody was used to increase the database of an existing Query by Humming system. The separation front-end based on the FChT was used to tackle the main shortcoming of QBH systems. The results are auspicious on the possibility to build the database without relying on a manual transcription of the melody nor need the information gathered from users consulting the database

Finally, two source separation techniques are described, one of them based on the FChT and the other based on the classical STFT using a sparse representation with structure. While they are completely different in the formulation, the F0gram works as a method to detect structure in a representation space where the signals of interest are represented in a sparse way.

The FChT has proven to be a powerful representation with diverse applications in the Music Information Retrieval field. The implementation is simple and can be easily combined with other techniques such as the CQT to be adapted to the characteristics of particular problems. The extension to nonlinear warps gives marginal improvements with a trade-off that suggests that the FChT as a low-level representation technique is close to a practical limit of the representation of the components of harmonic sounds in the applications considered.

The FChT captures the voiced components of a signal. In the case of speech, the separation of a voice based in the FChT can remove the voiced components but unvoiced components, which can be heard as a whispering voice, remain. As the voiced and unvoiced components go through the same resonance system, the next steps will consider to use the spectral envelope of the estimated harmonic signal in order to estimate and remove the unvoiced components.

The calculation of the F0gram applies a whitening of the spectrum as a robust measure of the presence of energy for different frequency components, with good results when nothing is known a priori about the spectral envelope of the signal.

As it was explored in the Score Informed Source Separation, a model of the envelope (trough GMM) can be useful to improve the detection. Future work will incorporate an envelope model in the representation of the F0gram when a training dataset of the source to be represented is available.

In this work, different MIR applications where approached based on the FChT as a low-level representation. Other MIR algorithms that are based on the STFT will be adapted to use the FChT and take advantage of the improved representation of harmonic signals.

Bibliography

[1] L. Cohen. Time-frequency analysis: Theory and applications. Upper Saddle River, NJ, USA, 1995. Prentice-Hall, Inc. ISBN 0-13-594532-1.

[2] J. C. Brown. Calculation of a constant Q spectral transform. *JASA*, 89(1): 425–434, 1991.

[3] J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *JASA*, 92(5):2698–2701, 1992.

[4] K. L. Kashima and B. Mont-Reynaud. The bounded-Q approach to time-varying spectral analysis. Tech. Rep. STAN-M-28, Stanford University, 1985.

[5] F. C. C. B. Diniz, L. Kothe, S. L. Netto, and L. W. P. Biscainho. High-Selectivity Filter Banks for Spectral Analysis of Music Signals. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007. doi: 10.1155/2007/94704.

[6] M. Goto. A Real-time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-world Audio Signals. *Speech Communication (ISCA Journal)*, 43(4):311–329, 2004.

[7] K. Dressler. Sinusoidal Extraction Using and Efficient Implementation of a Multi-Resolution FFT. In *Proceedings of the DAFx-06*, Montreal, Canada, 2006.

[8] P. Flandrin. *Time-Frequency/Time-scale Analysis*. Wavelet Analysis and Its Applications. Academic Press, 1999.

[9] S. A. Abdallah and M. D. Plumbley. Unsupervised analysis of polyphonic music by sparse coding. *IEEE Trans. Neural Networks*, 17(1):179–196, 2006.

[10] L. Cohen. Time-frequency distributions: a review. *Proceedings of the IEEE*, 77(7):941 – 981, 1989.

[11] S. Mann and S. Haykin. The chirplet transform: physical considerations. *IEEE Transactions on Signal Processing*, 41(11):2745–2761, 1991.

[12] L. B. Almeida. The fractional fourier transform and time-frequency representations. *IEEE Transactions on Signal Processing*, 42(11):3084 – 3091, 1994.

[13] L. Weruaga and M. Képesi. The fan-chirp transform for nonstationary harmonic signals. *Signal Processing*, 87(6):1504–1522, 2007.

[14] M. Képesi and L. Weruaga. Adaptive chirp-based time-frequency analysis of speech signals. *Speech Communication*, 48(5):474–492, 2006.

[15] C. Kereliuk and P. Depalle. Improved hidden markov model partial tracking through time-frequency analysis. In *Proc. of the 11th Conference on Digital Audio Effects (DAFx-08), Espoo, Finland*, September 2008.

[16] M. Bartkowiak. Application of the fan-chirp transform to hybrid sinusoidal+noise modeling of polyphonic audio. In *16th European Signal Processing Conference*, 2008.

[17] P. Zhao, Z. Zhang, and X. Wu. Monaural speech separation based on multi-scale fan-chirp transform. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, pages 161–164, 2008.

[18] Robert Dunn and T.F. Quatieri. Sinewave analysis/synthesis based on the fan-chirp tranform. In *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, pages 247–250, Oct 2007. doi: 10.1109/ASPAA.2007.4393028.

[19] R.B. Dunn, T.F. Quatieri, and N. Malyska. Sinewave parameter estimation using the fast fan-chirp transform. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, pages 349–352, Oct 2009. doi: 10.1109/ASPAA.2009.5346528.

[20] P. Cancela, E. López, and M. Rocamora. Fan chirp transform for music representation. In *International Conference on Digital Audio Effects, 13th. DAFx-10. Graz, Austria*, 6-10 Sep 2010. URL http://iie.fing.edu.uy/publicaciones/2010/CLR10.

[21] L. Jure, E. López, M. Rocamora, P. Cancela, H. Spontón, and I. Irigaray. Pitch content visualization tools for music performance analysis. In *International Society for Music Information Retrieval Conference, 13th, Proceedings.*

ISMIR 2012. Porto, Portugal, pages 493–498. FEUP Edições, 2012. URL http://iie.fing.edu.uy/publicaciones/2012/JLRCSI12.

[22] J. S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[23] M. Rocamora and P. Cancela. Pitch tracking in polyphonic audio by clustering local fundamental frequency estimates. In *Brazilian AES Audio Engineering Congress, 9th. São Paulo, Brazil*, may, 17–19 2011. URL http://iie.fing.edu.uy/publicaciones/2011/RC11.

[24] M. Rocamora, P. Cancela, and A. Pardo. Query by humming: Automatically building the database from music recordings. *Pattern Recognition Letters*, 36(0):272 – 280, 2014. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/j.patrec.2013.04.006. URL http://www.sciencedirect.com/science/article/pii/S0167865513001566.

[25] P. Sprechmann, P. Cancela, and G. Sapiro. Gaussian mixture models for score-informed instrument separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 49–52, March 2012. doi: 10.1109/ICASSP.2012.6287814.

[26] P. Cancela, M. Rocamora, and E. López. An efficient multi-resolution spectral transform for music analysis. In *International Society for Music Information Retrieval Conference, 10th. ISMIR 2009. Kobe, Japan*, pages 309–314, 2009.

[27] J. S. Prater and C. M. Loeffler. Analysis and design of periodically time-varying IIR filters, with applications to transmultiplexing. *IEEE Transactions on Signal Processing*, 40(11):2715–2725, 1992.

[28] P. Cancela. Tracking melody in polyphonic audio. In *Music Information Retrieval Evaluation eXchange*, 2008.

[29] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, October 2002.

[30] I. F. Apolinário, L. W. P. Biscainho, M. Rocamora, and P. Cancela. Fan chirp transform with nonlinear time warping. In *Brazilian AES Audio Engineering Congress, 13th. São Paulo, Brazil*, pages 62–68, 2015. URL http://iie.fing.edu.uy/publicaciones/2015/ABRC15.

[31] I. Bent and A. Pople. "Analysis". *Grove Music Online*. Accessed June 16, 2012.

[32] R. Cogan. *New images of musical sound*. Harvard University Press. Cambridge, Massachusetts, 1985.

[33] T. Licata, editor. *Electroacoustic Music - Analytical Perspectives*. Greenwood Press, 2002.

[34] D. Leech-Wilkinson. *The Changing Sound of Music: Approaches to Studying Recorded Musical Performance*. Published online, London: CHARM, 2009.

[35] A. Krishnaswamy. Pitch measurements versus perception of south indian classical music. In *Proc. of the Stockholm Music Acoustics Conf., Sweden, Aug.*, 2003.

[36] J. Durrieu, B. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *Selected Topics in Signal Processing, IEEE Journal of*, 5(6):1180 –1191, Oct. 2011. ISSN 1932-4553.

[37] A. Klapuri. A method for visualizing the pitch content of polyphonic music signals. In *10th Int. Society for Music Information Retrieval Conf., Japan*, 2009.

[38] L. Litova-Nikolova. *Bulgarian folk music*. Bulgarian academic monographs. Marin Drinov Academic Pub. House, 2004.

[39] S. Petrov, M. Manolova, and D. Buchanan. "Bulgaria". *Grove Music Online*. Accessed April 5, 2012.

[40] unesco. Musics & musicians of the world: Bulgaria. AUVIDIS/UNESCO, 1983.

[41] Ha Nguyen and Luis Weruaga. Time-frequency analysis of vietnamese speech inspired on chirp auditory selectivity. In *PRICAI 2008: Trends in Artificial Intelligence, 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, December 15-19, 2008. Proceedings*, pages 284–295, 2008.

[42] M. Rocamora and P. Cancela. Pitch tracking in polyphonic audio by clustering local fundamental frequency estimates. In *Brazilian AES Audio Engineering Congress, 9th. São Paulo, Brazil*, may, 17–19 2011. URL http://iie.fing.edu.uy/publicaciones/2011/RC11.

[43] A. de CheveignÃ©. Multiple F0 estimation. In D. Wang and G. Brown, editors, *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, pages 45–79. IEEE / Wiley, 2006.

[44] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, aug. 2000. ISSN 0162-8828. doi: 10.1109/34.868688.

[45] F. R. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *JMLR*, 7:1963–2001, 2006.

[46] M. Lagrange, L. G. Martins, J. Murdoch, and G. Tzanetakis. Normalized cuts for predominant melodic source separation. *IEEE Trans. on ASLP*, 16 (2):278 –290, feb. 2008. ISSN 1558-7916. doi: 10.1109/TASL.2007.909260.

[47] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007. ISSN 0960-3174. URL http://dx.doi.org/10.1007/s11222-007-9033-z.

[48] A. K. Jain. Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[49] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in NIPS*, pages 1601–1608. MIT Press, 2004.

[50] P. Cancela, E. López, and M. Rocamora. Fan chirp transform for music representation. In *13th Int. Conf. on Digital Audio Effects, Austria*, sep. 2010. URL http://iie.fing.edu.uy/publicaciones/2010/CLR10.

[51] M.R. Every and J.E. Szymanski. A spectral-filtering approach to music signal separation. In *DAFx*, 2004.

[52] J. Ganseman, P. Scheunders, G. J. Mysore, and J. S. Abel. Evaluation of a score-informed source separation system. In *ISMIR*, pages 219–224, 2010.

[53] R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. In *ICASSP*, May 2011.

[54] D.D. Lee and H.S. Seung. Learning parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[55] S. K. Tjoa, M. C. Stamm, W. Sabrina Lin, and K. J. Ray Liu. Harmonic variable-size dictionary learning for music source separation. In *ICASSP*, pages 413–416, Dallas, TX, mar 2010.

[56] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. Audio, Speech & Lang. Process.*, 18:538–549, 2010. ISSN 1063-6676.

[57] J. L. Durrieu, N. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *IEEE Journal of Sel. Topics in Signal Process.*, 5(6):1180 –1191, oct. 2011.

[58] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *CoRR*, abs/1006.3056, 2010. http://arxiv.org/abs/1006.3056.

[59] R. Jenatton., J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523, 2009.

[60] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech & Lang. Process.*, 14: 1462–1469, 2006.

[61] D. W. Griffin and J. S. Lim. Signal estimation from modified short-time fourier transform. *IEEE Trans. Acoustics, Speech and Sig. Proc*, pages 236–243, 1984.

[62] J. L. Roux and E. Vincent. Consistent wiener filtering for audio source separation. *IEEE Signal Process. Lett.*, 20(3):217–220, 2013. URL http://dblp.uni-trier.de/db/journals/spl/spl20.html#RouxV13.

[63] M. Müller, M. Goto, and M. Schedl, editors. *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2012. ISBN 978-3-939897-37-8.

[64] M. Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. ISBN 3540740473.

[65] A. Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. A John Wiley & Sons, Inc., publication. John Wiley & Sons, 2012. ISBN 9781118266823.

[66] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, New York, 2006. ISBN 0-387-30667-6.

[67] Ò. Celma. *Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, 2010. ISBN 978-3-642-13286-5.

[68] Y. Li and D. Wang. Singing voice separation from monaural recordings. In *Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006, Victoria, Canada, 8-12 October*, pages 176–179, 2006.

[69] M. Ryynänen and A. Klapuri. Transcription of the singing melody in polyphonic music. In *Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006, Victoria, Canada, 8-12 October*, pages 222–227, 2006.

[70] B. Pardo, J. Shifrin, and W. Birmingham. Name that tune: A pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology*, 55(4):283–300, 2003.

[71] B. Pardo, D. Little, R. Jiang, H. Livni, and J. Han. The vocalsearch music search engine. In *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital libraries*, JCDL '08, pages 430–430, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-998-2.

[72] J. Song, S. Y. Bae, and K. Yoon. Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System. In *Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR 2002, Paris, France, October 13-17*, pages 133–139, 2002.

[73] A. Duda, A. Nürnberger, and S. Stober. Towards query by singing/humming on audio databases. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23-27*, pages 331–334, 2007. ISBN 978-3-85403-218-2.

[74] M. Ryynänen and A. Klapuri. Query by Humming of MIDI and Audio Using Locality Sensitive Hashing. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, USA, March 30 - April 4*, pages 2249–2252, 2008.

[75] J. Salamon, J. Serrà, and E. Gómez. Tonal representations for music retrieval: From version identification to query-by-humming. *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, page In Press, 2013.

[76] E. López and M. Rocamora. Tararira: Query by singing system. In *The Second Annual Music Information Retrieval Evaluation eXchange (MIREX 2006), Abstract Collection*, pages 80–83. The International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL), Graduate School of Library and Information Science University of Illinois at Urbana-Champaign, 2006. Extended abstract.

[77] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith. Query by humming: musical information retrieval in an audio database. In *Proceedings of the third ACM international conference on Multimedia*, MULTIMEDIA '95, pages 231–236, New York, NY, USA, 1995. ACM. ISBN 0-89791-751-0.

[78] R.J. McNab, A. Lloyd Smith, I.H. Witten, C.L. Henderson, and S.J. Cunningham. Towards the digital music library: tune retrieval from acoustic input. In *Proceedings of the first ACM international conference on Digital libraries*, DL '96, pages 11–18, New York, NY, USA, 1996. ACM. ISBN 0-89791-830-4.

[79] N. Hu and R.B. Dannenberg. A comparison of melodic database retrieval techniques using sung queries. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, JCDL '02, pages 301–307, New York, NY, USA, 2002. ACM. ISBN 1-58113-513-0.

[80] Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 181–192, New York, NY, USA, 2003. ACM. ISBN 1-58113-634-X.

[81] K. Lemström. *String Matching Techinques for Music Retrieval.* PhD thesis, Department of Computer Science, University of Helsinki, Finland, 2000.

[82] J. Salamon and M. Rohrmeier. A quantitative evaluation of a two stage retrieval approach for a melodic query by example system. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, Japan, October 26-30*, pages 255–260, 2009.

[83] D. J. Levitin. Memory for musical attributes. In Perry R. Cook, editor, *Music, cognition, and computerized sound*, pages 209–227. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-03256-2.

[84] M. Rocamora and A. Pardo. Separation and classification of harmonic sounds for singing voice detection. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441 of *Lecture Notes in Computer Science*, pages 707–714. Springer, 2012.

[85] D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/.

[86] J. Sundberg. *The science of the singing voice*. De Kalb, Il., Northern Illinois University Press, 1987.

[87] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[88] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification. *Department of Computer Science, National Taiwan University*, 2007. Online web resource: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[89] B. Pardo and W. P. Birmingham. Query by humming: How good can it get? In *Workshop on the Evaluation of Music Information Retrieval Systems at SIGIR 2003, 1st August, Toronto, Canada*, pages 107–109, 2003.