Robust Temporal and Spectral Modeling for Query By Melody

by

Shai Shalev-Shwartz

Submitted to the School of Computer Science & Engineering
in partial fulfillment of the requirements for the degree of

Master of Computer Science

at the

HEBREW UNIVERSITY IN JERUSALEM

July 2002

This work was carried out under the supervision of
Dr. Yoram Singer and Dr. Shlomo Dubnov

Acknowledgments

I would like to thank my supervisor Dr. Yoram Singer for his given help all along the way.
I would also like to thank Dr. Shlomo Dubnov from the department of communication engineering in Ben-Gurion university for leading me through the paths of digital music processing.
In addition, I would like to thank Dr. Nir Friedman for his valuable suggestions.
I am grateful to Efrat Egozi and Leo Kontorovitch for their useful comments on the manuscript.
I want to thank the members of the machine learning lab for being a supporting company.
And last but not least, I would like to thank my wife Moria Shalev-Koffman for her support during the accomplishment of this work and also for her help in creating the queries used in the experiments.

Shai Shalev-Shwartz

Jerusalem, Israel
July 2002

Contents

List of Figures

Robust Temporal and Spectral Modeling for Query By Melody

by

Shai Shalev-Shwartz

Submitted to the School of Computer Science & Engineering
on July 31, 2002, in partial fulfillment of the
requirements for the degree of
Master of Computer Science

Abstract

Query by melody is the problem of retrieving musical performances from melodies. Retrieval of real performances is complicated due to the large number of variations in performing a melody and the presence of colored accompaniment noise. We describe a simple yet effective probabilistic model for this task. We describe a generative model that is rich enough to capture the spectral and temporal variations of musical performances and allows for tractable melody retrieval. While most of previous studies on music retrieval from melodies were performed with either symbolic (e.g. MIDI) data or with *monophonic* (single instrument) performances, we performed experiments in retrieving live and studio recordings of operas that contain a leading vocalist and rich instrumental accompaniment. Our results show that the probabilistic approach we propose is effective and can be scaled to massive datasets.

Thesis Supervisor: Dr. Yoram Singer and Dr. Shlomo Dubnov

Chapter 1

Introduction

A natural way for searching a musical audio database for a song is to look for a short audio segment containing a melody from the song. Most of the existing systems are based on textual information, such as the title of the song and the name of the composer. However, people often do not remember the name of the composer and the song's title but can easily recall fragments from the soloist's melody.

The task of *query by melody* attempts to automate the music retrieval task. It was first discussed in the context of query by humming [15, 19, 20, 21]. These works focus on converting hummed melodies into symbolic MIDI format (MIDI is an acronym for Musical Instrument Digital Interface. It is a symbolic format for representing music). Once the query is converted into a symbolic format the challenge is to search for musical performances that approximately match the query. Most of the research so far has been conducted with music stored in MIDI format [18] or in *monophonic* (i.e. single vocal or instrument) recordings (see for instance [12, 9] and the references therein). In this work, we suggest a method for query by melody where the query is posed in symbolic form as a monophonic melody and the database consists of real polyphonic recordings.

When dealing with real polyphonic recordings we need to address several complicating factors. Ideally, melodies can be represented as sequences of notes, each is a pair of frequency and temporal duration. In real recordings two major sources of difficulty arise. The first is the high variability of the actual durations of notes. A melody can be performed faster or slower than the one dictated by the musical score. This type of variation is often referred to as tempo variability. Furthermore, the tempo can vary within a single performance. For instance, a performance can start with a slow tempo which gradually increases. The second complicating factor is the high variability of the spectrum due to many factors such as differences in tone colors (timbre) of different singers/instruments, the intentional variation by the leading vocalists (e.g. vibrato and dynamics) and by "spectral masking" of the leading vocal by the accompanying vocals and orchestra.

We propose to tackle these difficulties by using a generative probabilistic approach that models the temporal and spectral variations. We associate each note with a hidden tempo variable. The tempo variables capture the temporal variations in the durations of notes. To enable efficient computation, the hidden tempo sequence is modeled as a first order Markov process. In addition, we describe a simple probabilistic spectral distribution model that is robust to the masking noise of the accompanying instruments and singers. This spectral distribution model is a variant of the harmonic likelihood model for pitch detection [29]. Combining the temporal and spectral probabilistic components, we obtain a joint model which can be considered as a dynamic Bayesian network [10]. This representation enables efficient alignment and retrieval using dynamic programming.

This probabilistic approach is related to several recent works that employ Hidden Markov Models (HMM) for music processing. Raphael [26] uses melody information (pitches and durations of notes) in building an HMM for a score following application. A similar approach is used by Durey and Clements [12] who use the pitch information of notes for building HMMs for melody retrieval. However, both approaches were designed for and evaluated on monophonic music databases. Most work on polyphonic music processing addressed tasks such as music segmentation into textures [6], polyphonic pitch tracking [31], and genre classification [30, 14]. We believe that the approach we describe in this paper is a step toward an effective retrieval procedure for massive musical datasets.

Chapter 2

Basic Concepts

A system for query by melody aims to connect between a melody and its performances. The substance of this connection is statistical. In the following, we explain basic concepts regarding two issues. In Sec. (2.1) we describe how we represent and analyze real performances using a computer. In Sec. (2.2) we review several statistical concepts and methods.

2.1 Digital musical audio signals

Basically, a musical audio signal is a sound wave, meaning, changes in the air density as a function of time. A microphone can convert a sound wave into the form of electric oscillations. Both of the above signals are analog - these signals are continuous functions from $\mathbb{R}$ to $\mathbb{R}$. A computer is a digital machine. Thus, in order to represent and analyze an analog signal using a computer, the signal has to be digitized. In this section we briefly introduce the basic concepts and methods of the theory of digital signal processing. Naturally, we focus on the aspects relevant to musical acoustic signals. After we lay the groundwork of basic digital signal processing, we describe the source-filter model of speech (and audio) production. For a more profound presentation see [27, 28, 22, 24, 16, 13, 7].

2.1.1 Analog and Digital Signals

An analog signal is a continuous value function of continuous time. A digital signal is a discrete value function of discrete time. Formally, let $\mathbb{R}$ denote the real numbers, let $\mathbb{C}$ denote the complex numbers and let $\mathbb{Z}$ denote the whole numbers. Let $\mathcal{Q}$ denote a finite set of rational numbers (usually, $|\mathcal{Q}| = 2^B$, where B is the number of bits we use for a single number). An analog signal $x_a(t)$ is a function from $\mathbb{R}$ to $\mathbb{C}$. Similarly,

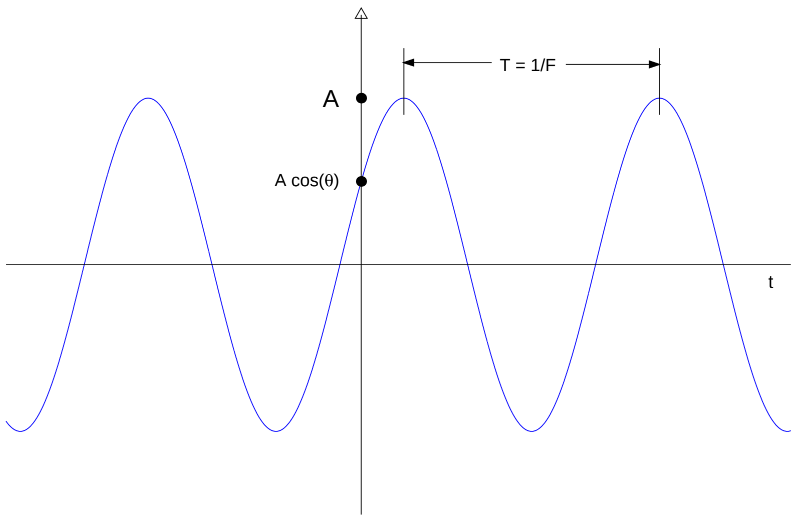

Figure 2-1: Example of an analog sinusoidal signal.

an analog signal of real values is a function from $\mathbb{R}$ to $\mathbb{R}$. A digital signal $x_d(n)$ is a function from $\mathbb{Z}$ to $\mathcal{Q}$. We use subscript $_a$ for representing analog signal and subscript $_d$ for representing digital signal. In the following, when it will be clear from the context, we will omit the subscript. A signal with argument t will denote an analog signal and a signal with argument n will denote a digital signal.

2.1.2 The concept of Frequency

Frequency is closely related to a specific type of periodic motion called harmonic oscillation, which is described by sinusoidal functions. A simple harmonic oscillation is mathematically described by the following analog sinusoidal signal:

$$x_a(t) = A\cos(2\pi Ft + \theta), \quad t \in \mathbb{R}$$

This signal is fully characterized by three parameters: A is the amplitude of the sinusoid, F is the *frequency* in cycles per second or hertz (Hz), and θ is the phase in radians. In Fig. (2-1) we illustrate these parameters.

It can easily be shown, that the sinusoidal signal is periodic. We denote the fundamental period of the sinusoidal signal by $T = \frac{1}{F}$. The concept of frequency is directly related to the concept of time. Actually, it has the dimension of inverse time.

The relationship we have described for sinusoidal signals carry over to the class of complex exponential signals,

$$x_a(t) = Ae^{i(2\pi Ft+\theta)},$$

where $i = \sqrt{-1}$. This can easily be seen by expressing the signals in terms of sinusoids using the Euler identity

$$e^{i\theta} = \cos\theta + i\sin\theta$$

According to Fourier (1768-1830), most signals of practical interest can be decomposed into a sum of complex exponential signals. Therefore, we can represent a signal in the time domain (as a function of time) or in the frequency domain (as a function of frequency). The representation of a signal using the second method is called the *spectrum* of the signal. The absolute value of the spectrum is called the *magnitude spectrum.* We use lower case letters for representing a signal in the time domain and upper case letters for representing a signal in the frequency domain. Similar to our notation in the time domain, a spectrum with argument f will denote continuous spectrum and a spectrum with argument k will denote discrete spectrum. We will explain how to convert a signal from the time domain to the frequency domain and vice versa in Sec. (2.1.4).

2.1.3 Analog to Digital Conversion

The process of *analog-to-digital* (A/D) conversion consists of two steps: sampling and quantization.

Sampling

We sample the analog signal periodically every T_s seconds. T_s is the *sampling period.* Thus, the n value of the sampled signal equals the value of the analog signal at time nT_s,

$$x_d(n) = x_a(nT_s) \quad .$$

The *sampling frequency* is the number of samples in one second. We denote the *sampling frequency* by F_s. Clearly,

$$F_s = \frac{1}{T_s} \quad .$$

The measure unit is samples per second. The *sampling frequency* is an upper bound for the frequencies that the sampled signal can express. In order to clarify this, let us observe two analog sinusoidal signals:

$$x_a(t) = \cos(2\pi f_x t) \quad , \quad y_a(t) = \cos(2\pi f_y t) \quad .$$

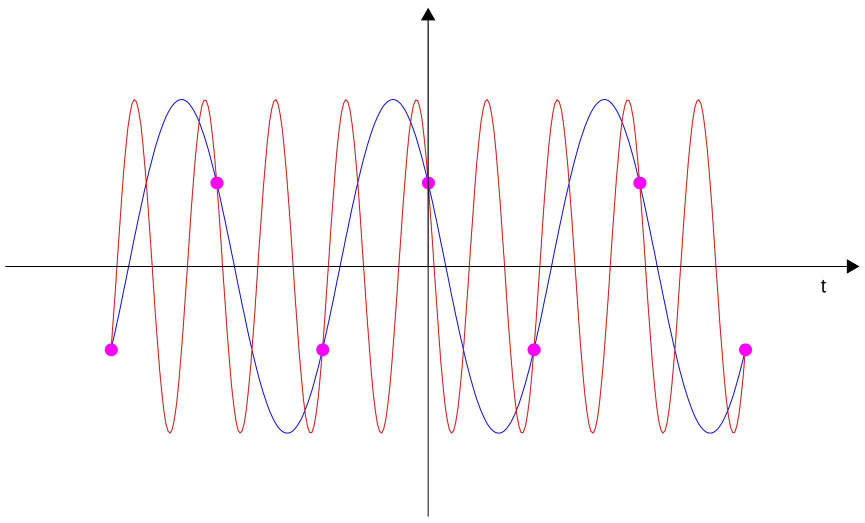

Figure 2-2: Example of aliasing.

We sample the signals using *sampling frequency* F_s. We choose F_s such that $F_s = f_y - f_x$ (without loss of generality $f_y > f_x$). We denote the resulting digital signals by $x_d(n)$ and $y_d(n)$ respectively. We can easily prove that x_d equals y_d:

$$\begin{aligned}
y_d(n) &= y_a(\frac{n}{F_s}) \\
&= \cos(2\pi \frac{f_y}{F_s} n) = \cos(2\pi \frac{F_s + f_x}{F_s} n) \\
&= \cos(2\pi n + 2\pi \frac{f_x}{F_s} n) = \cos(2\pi \frac{f_x}{F_s} n) \\
&= x_a(\frac{n}{F_s}) = x_d(n) \quad .
\end{aligned}$$

We illustrate this phenomenon in Fig. (2-2). Therefore, frequencies that are higher than the *sampling frequency* are introduced in the sampled signal as lower frequencies. This phenomenon is referred to as *aliasing*. The above discussion leads to the sampling theorem:

Theorem 1 (Sampling Principle) *An analog signal $x_a(t)$ can be reconstructed from its sampled signal $x_d(n) = x_a(nT_s)$ if the sampling frequency $F_s = \frac{1}{T_s}$ is greater than twice its highest frequency - F_0. Otherwise aliasing would result in $x_d(n)$. The sampling rate of $2F_0$ is called the Nyquist rate.*

Quantization

In order to represent each value of the sampled signal using B bits, we round each value to yield a discrete binary number. We refer to the rounding errors as *quanti-*

zation noise. Usually, we use a normalized 32 bit floating point format with values ranging from -1.0 to 1.0, and thus the *quantization noise* is neglected.

2.1.4 Fourier Transform

The French mathematician, Jean Baptiste Joseph Fourier (1768-1830), developed several mathematical formulas for converting signals from the time domain to the frequency domain and vice versa. We denote the conversion from the time domain to the frequency domain by *analysis* equation, and the opposite conversion by *synthesis* equation. The conversion formula is dependent on two aspects of the signal. Periodic (with period F_p) vs. aperiodic (with finite-energy, see Sec. (2.1.5)) signals and continuous-time vs. discrete-time signals. In table (2.1) we describe the *analysis* and *synthesis* equations for the different cases. The only class of signals we can store in a computer is the periodic discrete-time signals (we only need to store the N values of one period of the signal). We also refer to this class of signals as finite-duration discrete-time signals. The *analysis* equation for this class is often referred to as the *Discrete Fourier Transform* (DFT). Similarly, the *synthesis* equation for this class is often referred to as the *Inverse Discrete Fourier Transform* (IDFT).

The spectrum of a periodic discrete-time signal with a period N also form a periodic sequence with a period N:

$$X(k+N) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-i2\pi\frac{(k+N)}{N}n} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-i2\pi\frac{k}{N}n} = X(k) \quad .$$

If we use a sampling frequency F_s, the range $0 \leq k \leq N-1$ corresponds to the frequency range $0 \leq f \leq F_s$. Therefore, $X(k)$ is a sampling of the spectrum range $[0, F_s]$.

The complexity of the straightforward implementation of the DFT is $O(N^2)$. By taking advantage of the periodicity of the analysing sinusoids $e^{-i2\pi\frac{k}{N}n}$ and applying the divide and conquer principle of splitting the problem into successively smaller subproblems, a variety of more efficient algorithms of complexity $O(N \log N)$ have been developed which came to be known collectively as the *Fast Fourier Transform* (FFT). They generally require that N be a power of two. This requirement is not critical, since we can add zeros to the end of the finite signal. It can be shown that zero padding a signal only increases the sampling rate of the frequency.

		Analysis	Synthesis
periodic	continuous	$X(k) = \frac{1}{T_p} \int_{T_p} x(t) e^{-i2\pi k F_p t} dt$	$x(t) = \sum_{k=-\infty}^{\infty} X(k) e^{i2\pi k F_p t}$
	discrete	$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-i2\pi \frac{k}{N} n}$	$x(n) = \sum_{n=0}^{N-1} X(k) e^{i2\pi \frac{k}{N} n}$
aperiodic	continuous	$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi f t} dt$	$x(t) = \int_{-\infty}^{\infty} X(f) e^{i2\pi f t} df$
	discrete	$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-i2\pi f n}$	$x(n) = \int_{-\pi}^{\pi} X(f) e^{i2\pi f n} df$

Table 2.1: Fourier transforms.

2.1.5 Energy

Definition 1 (Energy.) *The energy of a discrete-time signal $x(n)$ is defined as*

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2$$

A signal $x(n)$ has finite energy iff $E < \infty$.

Usually, we are interested in the ratio between two energy values. The *decibel* unit is the common way for comparing the relative energies of two signals.

Definition 2 (Decibel.) *The decibel is defined as*

$$1\ dB \equiv 10 \log_{10} \frac{E_1}{E_0}$$

where E_0 is the reference energy.

We can also calculate the energy of the spectrum. However, Parseval's Theorem states that the two calculations lead to the same value. We state the theorem for finite-duration discrete-time signals.

Theorem 2 (Parseval.)

$$\sum_{n=0}^{N} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N} |X(k)|^2$$

The above equality can be easily verified using the idea that orthogonal linear transformation do not change the norm of a vector, and the DFT is indeed an orthogonal linear transformation.

2.1.6 Linear Shift-Invariant Systems

A *system* $T[\cdot]$ is a function from the signals space to the signals space. We also use the term *filter* for describing systems. We will mainly refer to systems that operate on digital signals. We refer to the argument of this function as the *input* of the system and to the value of the function as the *output* of the system. We usually limit ourselves to a specific class of systems - the *Linear Shift-Invariant* (LSI) Systems.

Definition 3 (Linear Systems.) *A discrete system $T[\cdot]$ is* linear *if and only if $T[\cdot]$ satisfies the principle of superposition, namely, for all two digital signals $x_1(n), x_2(n)$ and for all all two superposition coefficients $a_1, a_2 \in \mathbb{C}$,*

$$T[a_1 x_1(\cdot) + a_2 x_2(\cdot)] = a_1 T[x_1(\cdot)] + a_2 T[x_2(\cdot)] \quad .$$

Definition 4 (Shift-Invariant (SI) Systems.) *Let $S_k[\cdot]$ be a system that shifts a signal k samples, namely, if $y(\cdot) = S_k[x(\cdot)]$ then $\forall n, \quad y(n) = x(n-k)$. A discrete system $T[\cdot]$ is* shift-invariant *if and only if shifting the input of the system is equivalent to shifting the output of the system*

$$T[S_k[x(\cdot)]] = S_k[T[x(\cdot)]] \quad .$$

LSI systems can be fully represented using their response due to a unit sample at time 0. We denote this input signal by

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & otherwise \end{cases} \quad .$$

We denote the output of a system due to the $\delta(n)$ input by the *impulse response* of the system

$$h(n) = T[\delta(n)] \quad .$$

We now show how to represent the output of a system $T[\cdot]$ using the impulse response. Let $x(n)$ be an input signal. We can write $x(n)$ using the natural basis of shifted unit sample signals

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) = \sum_{k=-\infty}^{\infty} x(k)S_k[\delta(n)] \quad .$$

Using the linear and shift-invariant properties, we can write the output of the system as

$$\begin{aligned} y(n) &= T[x(n)] \\ &= T\left[\sum_{k=-\infty}^{\infty} x(k)S_k[\delta(n)]\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)T\left[S_k[\delta(n)]\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)S_k\left[T[\delta(n)]\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad . \end{aligned} \tag{2.1}$$

The mathematical operation in Equ. (2.1) is called a linear *convolution* and is denoted by

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \equiv x(n) * h(n) \quad .$$

Thus, the impulse response fully describes the system. We can determine the output of the system by convolving the input signal with the impulse response of the system.

LSI Systems in the Frequency Domain

We saw that we can represent a LSI system using its impulse response, $h(n)$. The discrete-time Fourier transform of the impulse response is called the *Transfer Function* of the system and we denote it by

$$H(f) \equiv \sum_{n=-\infty}^{\infty} h(n) e^{-i2\pi f n} \quad .$$

Using the definitions of the Fourier transform and the convolution we can derive the following important property of the convolution:

$$\begin{aligned}
Y(f) &= \sum_{n=-\infty}^{\infty} y(n) e^{-i2\pi f n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(k) h(n-k) e^{-i2\pi f n} \\
&= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(k) h(n-k) e^{-i2\pi f(n-k)} e^{-i2\pi f k} \\
&= \sum_{k=-\infty}^{\infty} x(k) e^{-i2\pi f k} \sum_{n=-\infty}^{\infty} h(n-k) e^{-i2\pi f(n-k)} \\
&= \sum_{k=-\infty}^{\infty} x(k) e^{-i2\pi f k} \sum_{m=-\infty}^{\infty} h(m) e^{-i2\pi f(m)} \\
&= X(f) H(f)
\end{aligned}$$

A convolution in the time domain is equivalent to multiplication in the frequency domain. Therefore, the spectrum of the output at f only depends on the spectrum of the signal at f and on the value of the transfer function at f. It does not depend on the spectrum at other frequencies. In other words, the complex exponential signals are eigenvectors of any LSI system and the transfer function is the corresponding eigenvalues.

2.1.7 The Short Time Fourier Transform

The Fourier Transform is adequate for analyzing the spectrum of signals whose properties do not vary with time, that is for analyzing stationary signals. However, musical signals are not stationary. The properties of musical signals (amplitudes, frequency and phases) change with time. Therefore, we are interested in the spectrum of short segments of the signal. We calculate the time-dependent spectrum of the signal using the *Short-Time Fourier Transform.*

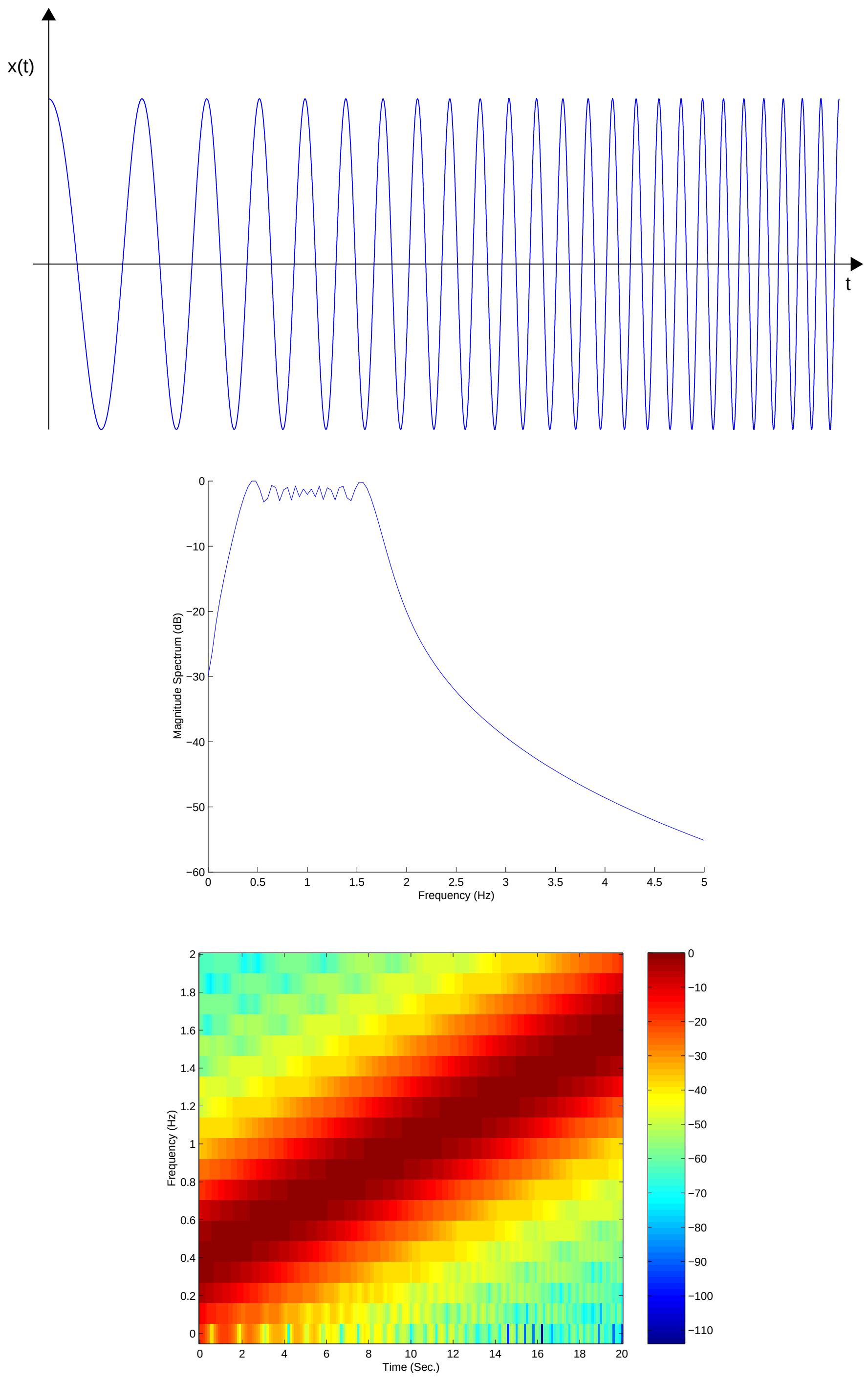

Figure 2-3: A chirp signal (top), its DFT analysis using dB scale (middle) and its STFT analysis presented by spectrogram (bottom).

Definition 5 (Short Time Fourier Transform.) *The* Short-Time Fourier Transform *(STFT) of a signal $x(n)$ is defined as*

$$X(n,f) = \sum_{m=-\infty}^{\infty} x(n+m)w(m)e^{-i2\pi fm} \quad , \tag{2.2}$$

where $w(n)$ is a window signal.

Equ. (2.2) can be interpreted as the DFT of the shifted signal $x(n+m)$, as viewed through the window $w(m)$. In the STFT, the one-dimensional sequence $x(n)$ is converted into a two dimensional function of time and frequency. In music signals, the time resolution for the STFT is usually between $10ms$ to $80ms$. We use the *spectrogram* for illustrating the magnitude of the STFT graphically. In Fig. (2-3) we compare the DFT and the STFT of a chirp signal. The frequency of the chirp signal changes linearly through time. The DFT analysis shows the mean frequency contents of the signal. The STFT analysis shows the changes of the frequency content through time.

Windowing

The primary purpose of the window $w(n)$ in the STFT is to limit the extent of the sequence to be transformed so that the spectral characteristics are approximately stationary over the duration of the window. The more rapidly the signal characteristics change, the shorter the window should be. On the other hand, a multiplication with a window blurs the spectrum of the signal. To understand this, recall that a convolution in the time domain is equivalent to multiplication in the frequency domain. Similarly, a multiplication in the time domain is equivalent to convolution in the frequency domain. Therefore, the original spectrum of the signal is convolved with the spectrum of the window. In Fig. (2-4) we show the spectrum of a rectangular window. This shape is often referred to as *sinc* function. Reduced resolution and leakage are the two primary effects on the spectrum as a result of applying a window to the signal. The resolution is influenced primarily by the main lobe of the *sinc*, while the degree of leakage depends on the relative amplitude of the main lobe and the side lobes. Both the main lobe width and the relative amplitude of the main lobe and the side lobes are dependent on the length of the window and on its shape. For a more profound presentation see [22, 24, 16].

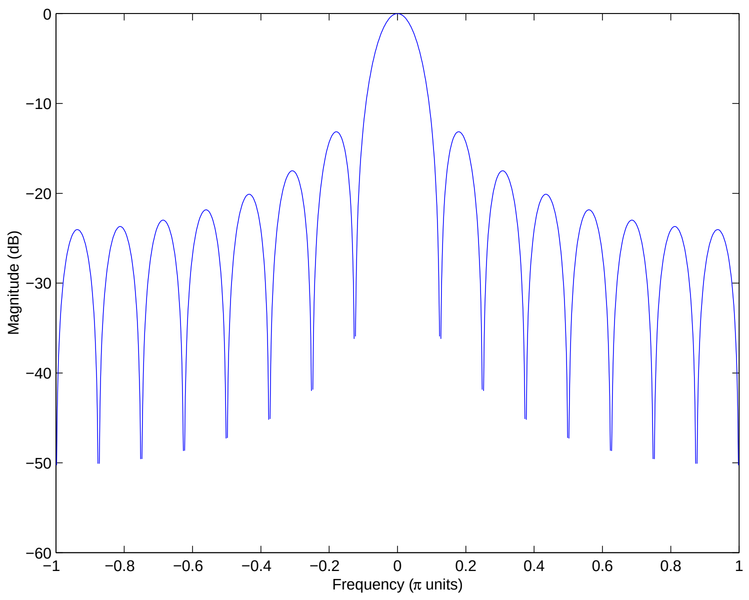

Figure 2-4: The spectrum of rectangular window.

2.1.8 Musical Sound Generation

In this section we briefly explain the physical bases of sound generation using traditional musical instruments. All traditional instruments have an oscillating element and some means of coupling the sound to the surrounding. For example, in the violin family, the oscillating element is a string. The string is coupled to a sounding board using a bridge (see Fig (2-5)). The sounding board receives mechanical motion from the string and provides better acoustical coupling to the air than can the string alone, with its very small surface area.

Using the signal processing terminology, the oscillating element can be viewed as a source signal $s(t)$ that passes through a system (e.g. in the violin family, the system depends on the sound board). We usually assume that the system is LSI, and thus is represented using its impulse response $h(t)$. We now explore the properties of the source signal. If we set a string into motion at its midpoint, it will vibrate at a fundamental frequency given by

$$f_0 = \frac{1}{2L}\sqrt{\frac{T}{m}} \quad ,$$

where L is the length of the string, T is its tension and m is its mass. This mode of vibration is called a standing wave. The string can also vibrates at multiplication of the fundamental frequency. These frequencies are known as harmonic frequencies, or merely harmonics. In the general case, the string can vibrate in all these frequencies at the same time. At any frequency other than a harmonic frequency, the interference of reflected and incident waves results in a resulting disturbance of the medium which is irregular and non-repeating. In Fig. (2-6) we illustrate the vibrating of the string at the harmonics.

Thus, the source $s(t)$ is a periodic signal that can be expressed using its fundamental frequency

$$s(t) = \sum_{h=1}^{\infty} sin(2\pi h f_0 t) \quad .$$

Using the definition of the Fourier transform for periodic continuous signals from Sec. (2.1.4), we can calculate the spectrum of the signal

$$\begin{aligned}
S(k) &= f_0 \int_{1/f_0} \left(\sum_{h=1}^{\infty} sin(2\pi h f_0 t) \right) e^{-i2\pi f_0 k t} dt \\
&= f_0 \int_{1/f_0} \left(\sum_{h=1}^{\infty} \left[\frac{e^{i2\pi h f_0 t} - e^{-i2\pi h f_0 t}}{2i} \right] \right) e^{-i2\pi f_0 k t} dt \\
&= \frac{f_0}{2i} \sum_{h=1}^{\infty} \int_{1/f_0} e^{i2\pi (h-k) f_0 t} dt - \int_{1/f_0} e^{-i2\pi (h+k) f_0 t} dt \\
&= \frac{1}{2i}
\end{aligned}$$

$$\Longrightarrow \qquad \forall k, \quad |S(k)| = constant \quad . \tag{2.3}$$

This kind of spectrum is called a *pulse train.*

As we mentioned, the source signal $s(t)$ is filtered using the system $h(t)$. Therefore, the musical signal is a convolution of the source signal with the impulse response of the system $h(t)$

$$x(t) = s(t) * h(t) \quad ,$$

and in the frequency domain,

$$X(f) = S(f)H(f) \quad .$$

From this analysis we conclude that $X(f)$ is a scaled pulse train with fundamental frequency f_0.

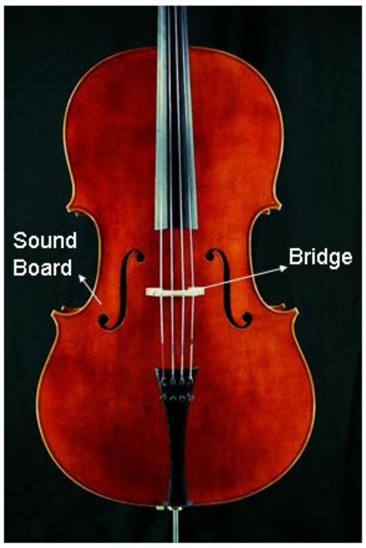

Figure 2-5: The elements of a cello.

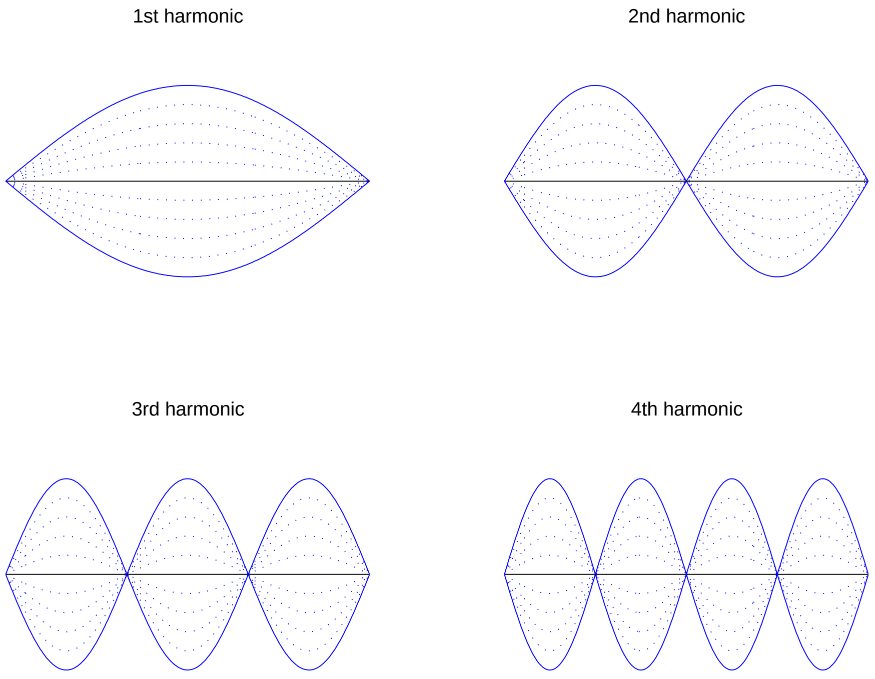

Figure 2-6: Vibration of a string at the harmonic frequencies.

2.2 Statistics

In this section we define the basic statistical tools we use for our retrieval system. For a more profound presentation see [8, 25, 10, 17].

2.2.1 Probability and Random variables

We define probability in terms of a *sample space* S, which is a set whose elements are called *elementary elements.* Each elementary element can be viewed as a possible outcome of an experiment. A *probability distribution* $Pr\{\cdot\}$ on a sample space S is a mapping from events of S to real numbers such that the following *probability axioms* are satisfied:

1. $Pr\{A\} \geq 0$ for any event A.
2. $Pr\{S\} = 1$.
3. $Pr\{A \cup B\} = Pr\{A\} + Pr\{B\}$ for any events A and B such that $A \cap B = \phi$.

A *random variable* X is a function from the sample space to the real numbers. It associates a real number with each outcome of an experiment, which allows us to work with the probability distribution induced on the resulting set of numbers. For a random variable X and a real number x, we define the event $X = x$ to be $\{s \in S : X(s) = x\}$; thus,

$$Pr(X = x) = \sum_{\{s \in S : X(s) = x\}} Pr\{s\} \quad .$$

2.2.2 Conditional probability

Sometimes we have some prior knowledge about the outcome of an experiment. For example, suppose that a friend has flipped two fair coins and has told you that at least one of the coins showed a head. What is the probability that both coins are heads? The information given eliminates the possibility of two tails.

Conditional probability formalizes the notion of having prior partial knowledge of the outcome of an experiment. The *conditional probability* of an event $X = x$ given that another event $Y = y$ has occurred is defined to be

$$Pr(X = x | Y = y) = \frac{Pr(X = x, Y = y)}{Pr(Y = y)} \quad . \tag{2.4}$$

The conditional probability of $X = x$ given $Y = y$ is the ratio of the probability of both events $X = x$ and $Y = y$ to the probability of event $Y = y$.

From the definition of conditional probability (Equ. (2.4)), it follows that for two events $X = x$ and $Y = y$

$$\begin{aligned} Pr(X = x, Y = y) &= Pr(Y = y)Pr(X = x|Y = y) \\ &= Pr(X = x)Pr(Y = y|X = x) \quad . \end{aligned}$$

Solving for $Pr(X = x|Y = y)$, we obtain

$$\begin{aligned} Pr(X = x|Y = y) &= \frac{Pr(X = x)Pr(Y = y|X = x)}{Pr(Y = y)} \\ &= \frac{Pr(X = x)Pr(Y = y|X = x)}{\sum_{\widetilde{x}} Pr(X = \widetilde{x})Pr(Y = y|X = \widetilde{x})} \quad , \end{aligned}$$

which is known as *Bayes's theorem.* The denominator is a normalizing factor.

2.2.3 Conditional Independencies and Bayesian Networks

Two random variables are *independent* if

$$\forall x, y \quad Pr(X = x, Y = y) = Pr(X = x)Pr(Y = y) \quad . \tag{2.5}$$

We also use the notation $Pr(X, Y) = Pr(X)Pr(Y)$ with upper case letter, to express the same equality as in Equ. (2.5). An equivalent definition is

$$Pr(X|Y) = Pr(X) \quad ,$$

which tells us that independence is also a notion of information. Knowing the value of Y tells us nothing about the value of X. In general, if $X_1, ..., X_n$ are independent random variables, then

$$Pr(X_1, ..., X_n) = \prod_{i=1}^{n} Pr(X_i) \quad ,$$

which allows us a compact representation of the joint distribution.

Unfortunately, most of random variables of interest are dependent on each other. Therefore, we define a more refined notion - *conditional independence.* It applies when two variables are independent given the value of another variable. Formally, variables X and Y are *conditionally independent*, given variable Z, if

$$Pr(X, Y|Z) = Pr(X|Z)Pr(Y|Z) \quad ,$$

that is, learning the values of Y does not change our prediction of X once we know the value of Z.

The conditional independencies between variables can be visualized using a special graph called a *Bayesian network structure.*

Definition 6 *A* Bayesian network structure *G is a directed acyclic graph whose nodes represent random variables $X_1, ..., X_n$. Let Pa_{X_i} denote the parents of X_i in G, and Nd_{X_i} denote the variables in the graph that are not descendants of X_i. Then G encodes the following set of conditional independence assumptions:*

$$\forall X_i, \quad Pr(X_i, Nd_{X_i}|Pa_{X_i}) = Pr(X_i|Pa_{X_i})Pr(Nd_{X_i}|Pa_{X_i}) \ ,$$

that is, X_i is independent of its non-descendants given its parents.

2.2.4 Markov models

A *discrete stochastic process* is an indexed sequence of random variables. In general, there can be an arbitrary dependence among the random variables. The process is characterized by the joint probability functions $Pr(X_1, X_2, ..., X_n)$ for $n = 1, 2, ...$.

Definition 7 *A discrete stochastic process is said to be* stationary *if the joint distribution of any subset of the sequence of random variables is invariant with respect to shifts in the time index, i.e.,*

$$\forall \ l, \quad Pr(X_1, ..., X_n) = Pr(X_{1+l}, ..., X_{n+l})$$

A simple example of a stochastic process with dependence is one in which each random variable depends on the one preceding it and is *conditionally* independent of all the other preceding random variables. Such a process is said to be a *first order Markov process.*

Definition 8 *A discrete stochastic process $X_1, X_2, ...$ is said to be a* first order Markov process *if for $n = 1, 2, ...$,*

$$Pr(X_{n+1}|X_n, X_{n-1}, ..., X_1) = Pr(X_{n+1}|X_n) \ .$$

In this case, the joint probabilty function can be factorized as

$$Pr(X_1, ..., X_n) = Pr(X_1)\prod_{i=2}^{n} Pr(X_i|X_{i-1}) \ .$$

Definition 9 *A first order Markov process is said to be* time invariant *if the conditional probability $Pr(X_{n+1}|X_n)$ does not depend on n, i.e., for $n = 1, 2, ...$*

$$Pr(X_{n+1}|X_n) = Pr(X_2|X_1) \quad .$$

We will assume that a Markov process is time-invariant unless otherwise stated.

Chapter 3

Problem Setting

In our setting, we are given a melody and our task is to retrieve musical performances containing the requested melody and to find its location within the retrieved performances. In this chapter we explain our setting formally. In Sec. (3.1) we formally define melody. We continue with a formal definition of a performance in Sec. (3.2). We also explain why the task of matching between a melody and a performance is difficult.

3.1 A musical melody

A melody is a sequence of notes where each note is a pair of a pitch value and a duration value.

3.1.1 Pitch

In this work, we associate pitch with fundamental frequency, although the two are not quite the same (see for example [13, 7]).

There exists a discrete set of all possible frequencies of notes. In Fig. (3-1) we show one octave of the keyboard spanning the notes from C to the C in the next octave. The frequency relation between notes with an interval of one octave is $(1 : 2)$. There are twelve intervals within one octave. The basic interval is called a semi-tone. In the well-tempered Western music tuning system, the frequency relation between consecutive semi tones is constant. Therefore, this relation is $(1 : \sqrt[12]{2})$. Furthermore, this relation does not depend on the musical scale. (For other tuning systems see [13, 7]). Thus, if we know the frequency of a reference note, we can calculate the frequency of other notes using the interval between this note and the reference note (i.e. the number of semi tones between the notes). Formally, let Λ

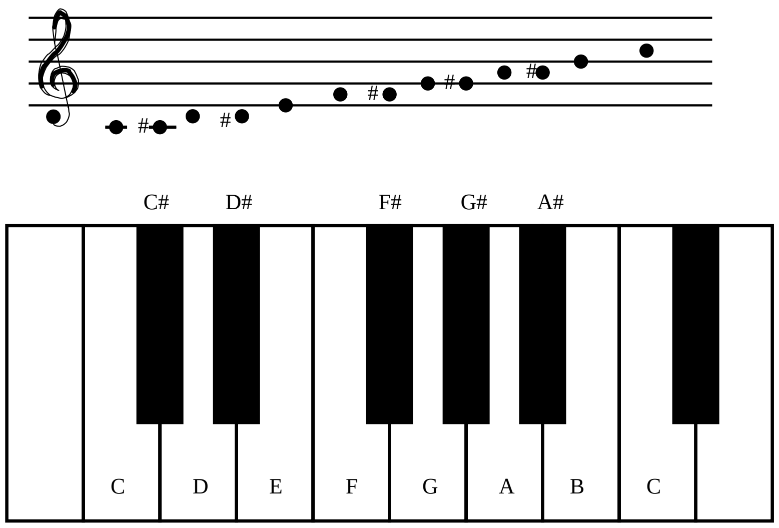

Figure 3-1: A twelve-to-the-octave scale.

denote the set of all possible frequencies of notes. In the well-tempered Western music tuning system, $\Lambda = \{f_{ref} \cdot 2^{s/12} | s \in \mathcal{Z}\}$, where $f_{ref} = 440Hz$.

Using the definition of Λ we can define a pitch of a note formally. Let $\mathbb{R}_+$ denote the positive real numbers. Let $f_l, f_h \in \mathcal{R}_+$ be frequency values (in Hz) and let $[f_l, f_h]$ be a *diapason.* A *diapason* of a singer (or an instrument) is the range of pitch frequencies that are in use by the singer (or by the instrument). For instance, a tenor singer typically employs a *diapason* of $[110Hz, 530Hz]$. Let $\Gamma = [f_{low}, f_{high}] \cap \Lambda$ denote all the possible pitches of notes in the *diapason.* A pitch of a note is an element in the set Γ.

3.1.2 Duration

Similarly to the pitch quantization, there is also a discrete set of all possible durations of notes. However, for simplicity, in this work we allow a duration to be any real positive number. That is, a duration of a note is a number in $\mathbb{R}_+$.

The above leads us to the formal definition of a melody.

Definition 10 (Melody) *A melody is described formally by a sequence of pitches,* $\boldsymbol{p} \in \Gamma^k$*, and a sequence of durations,* $\boldsymbol{d} \in {\mathcal{R}_+}^k$*, in a predefined time unit (e.g. seconds).*

3.2 A performance

Our goal is to retrieve melodies from audio signals representing real performances.

Rallentando	1.2 1.2 1.25 1.3 1.3
Accelerando	0.7 0.65 0.6 0.5 0.5

Table 3.1: Examples of scaling factor sequences: In the first sequence the scaling factors are gradually increasing and thus the tempo is decreasing ("Rallentando"). In the second example the scaling factors are decreasing and the tempo is increasing ("Accelerando").

Definition 11 (Performance) *A performance of a melody is a discrete time sampled audio signal, $\boldsymbol{o} = o_1, ..., o_T$.*

A performance is formally entirely defined given the melody: play or sing using pitch p_1 for the first d_1 seconds, then play or sing pitch p_2 for the next d_2 seconds, and so on and so forth. In reality, a melody does not impose a rigid framework. The actual frequency content of a given note varies with the type of instrument that is played and by the performer. Examples for such variations are the vibrato and timbre effects. The accompaniment also greatly influences the spectral distribution. While playing a note using pitch p, we are likely to see a local concentration of energy close to multiples of the frequency p in the power spectrum of the signal. However, there may be other spectral regions with high levels of energy. We will address this problem later on in this section. Another source of variation is local scaling of the durations of notes as instructed by the melody. The performer typically uses a tempo that scales the duration and moves from one tempo to another, thus using a different time scale to play the notes. Therefore, we also need to model the variation in the tempo which we describe now.

A tempo sequence is a sequence of scaling factors, $\boldsymbol{m} \in \mathcal{R}_+{}^k$. The actual duration of note i, denoted $\widetilde{d_i}$ is d_i scaled by m_i, $\widetilde{d_i} = d_i m_i$. Seemingly, allowing different scaling factors for the different notes adds a degree of freedom that makes the melody duration values redundant. However, a typical tempo sequence does not change rapidly and thus reflects most of the information of the original durations (up to a scaling factor). Table 3.1 shows two examples of tempo sequences. A pitch–duration–tempo triplet $(\boldsymbol{p}, \boldsymbol{d}, \boldsymbol{m})$ generates an actual pitch–duration pair $(\boldsymbol{p}, \widetilde{\boldsymbol{d}})$.

In order to describe the generation of the actual performance audio signal $\boldsymbol{o}$ from $(\boldsymbol{p}, \widetilde{\boldsymbol{d}})$ we introduce one more variable, $\boldsymbol{s} \in \mathcal{R}_+{}^k$ where s_i is the starting time (sample number) of note i in the performance. We define $s_i = 1 + \sum_{j=1}^{i-1} \widetilde{d_j}$ for $i = 1, \ldots, k+1$. Notes generate consecutive blocks of signal samples. Let $\overline{o}_i = (o_{s_i}, ..., o_{s_{i+1}-1})$ be the block of samples generated by note i.

The magnitude spectrum of $\overline{o}_i$ varies significantly from performance to performance, according to various factors such as the spectral envelope of the soloist and pitches of accompaniment instruments. Since our goal is to locate and retrieve a melody from a dataset that may contain thousands of performances, we resort to a very simple spectral model and do not explicitly model these variables. We use an approximation to the likelihood of a block spectrum given its pitch.

Chapter 4

From melody to signal: a generative model

To pose the problem in a probabilistic framework, we need to describe the likelihood of a performance given the melody, $P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d})$. We cast the tempo sequence $\boldsymbol{m}$ as a hidden random variable, thus the likelihood can be written as,

$$P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d}) = \sum_{\boldsymbol{m}} P(\boldsymbol{o}, \boldsymbol{m}|\boldsymbol{p}, \boldsymbol{d}) \quad . \tag{4.1}$$

For simplicity, we assume that the tempo sequence does not depend on the melody. While this assumption, naturally, does not always hold, we found empirically that these types of correlations can be ignored in short pieces of performances. With this assumption and the identity $\widetilde{\boldsymbol{d}} = \boldsymbol{dm}$, Equ. (4.1) can be rewritten as,

$$\begin{aligned} P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d}) &= \sum_{\boldsymbol{m}} P(\boldsymbol{m}|\boldsymbol{p}, \boldsymbol{d}) P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d}, \boldsymbol{m}) \\ &= \sum_{\boldsymbol{m}} P(\boldsymbol{m}) P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d}, \boldsymbol{m}) \\ &= \sum_{\boldsymbol{m}} P(\boldsymbol{m}) P(\boldsymbol{o}|\boldsymbol{p}, \widetilde{\boldsymbol{d}}) \quad . \end{aligned}$$

We illustrate the independencies between the variables in Fig. (4-1). We now need to specify the prior distribution over the tempo, $P(\boldsymbol{m})$, and the posterior distribution of the signal given the pitches and the actual durations of the notes $P(\boldsymbol{o}|\boldsymbol{p}, \widetilde{\boldsymbol{d}})$.

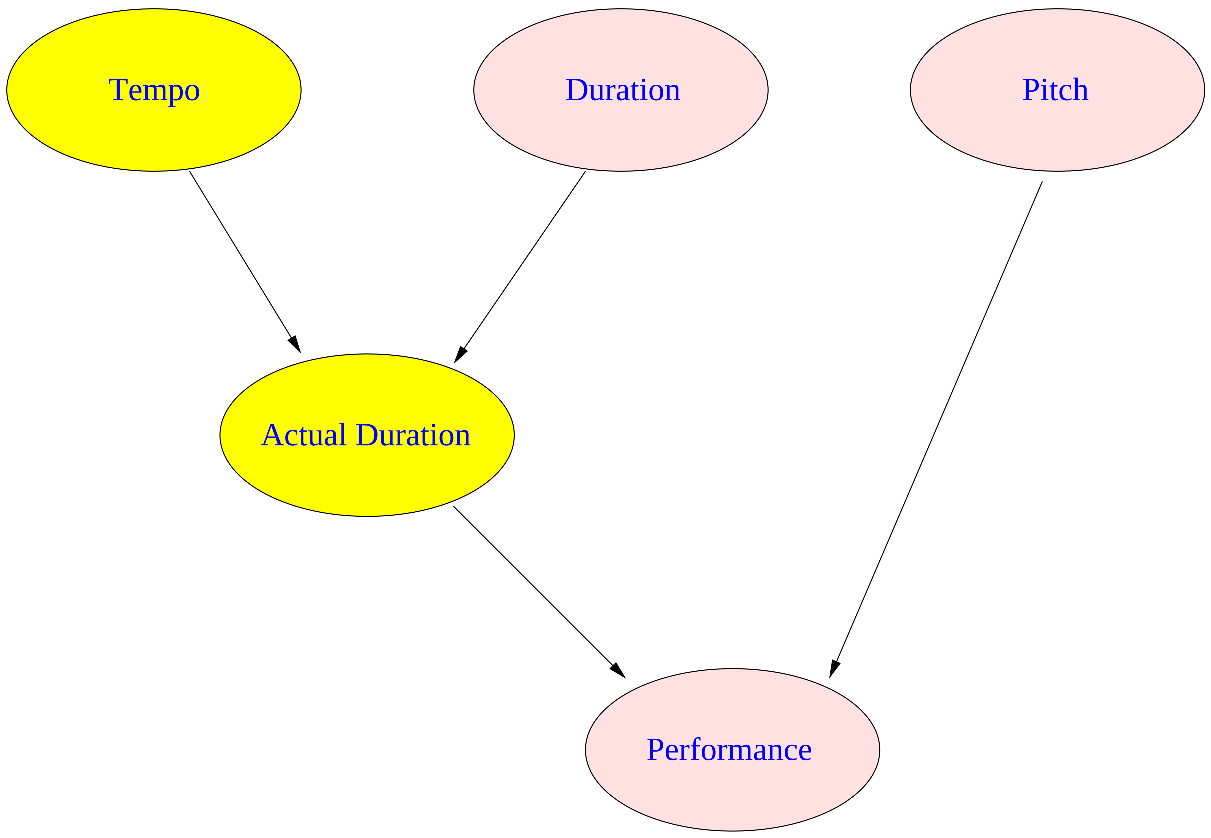

Figure 4-1: The independencies between the variables. The observed variables are pink colored and the hidden variables are yellow colored.

4.1 Tempo modeling

We chose to model the tempo sequence as a first order Markov process. As we see in the sequel this choice on one hand allows an efficient alignment and retrieval, and on the other hand, was found empirically to be rich enough. Therefore, the likelihood of $\boldsymbol{m}$ is given by,

$$P(\boldsymbol{m}) = P(m_1)\prod_{i=2}^{k} P(m_i|m_{i-1}) \quad .$$

We use the log-normal distribution to model the conditional probability $P(m_i|m_{i-1})$, that is $\log(m_i) \sim \mathcal{N}(\log(m_{i-1}), \rho)$, where ρ is a scaling parameter of the variance. The prior distribution of the first scaling factor $P(m_1)$ is also assumed to be log-normal around zero with variance ρ, $\log(m_i) \sim \mathcal{N}(0, \rho)$. In our experiments, the parameter ρ was determined manually according to musical knowledge. This parameter can also be learned from MIDI files. Note that for the log-normal distribution, the probability to change the tempo from value x to value λx is equal to the probability to change the tempo from value x to value $\frac{x}{\lambda}$,

$$\begin{aligned}
P(m_i = \lambda x | m_{i-1} = x) &= \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{1}{2\rho}(\log(\lambda x) - log(x))^2} \\
&= \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{1}{2\rho}(\log(\lambda))^2} \\
&= \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{1}{2\rho}(-\log(\lambda))^2} \\
&= \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{1}{2\rho}(\log(\frac{x}{\lambda x}))^2} \\
&= \frac{1}{\sqrt{2\pi\rho}} e^{-\frac{1}{2\rho}(\log(\frac{x}{\lambda}) - log(x))^2} \\
&= P(m_i = \frac{x}{\lambda} | m_{i-1} = x) \quad .
\end{aligned}$$

4.2 Spectral Distribution Model

In this section we describe our spectral distribution model. There exist quite a few models for the spectral distribution of singing voices and harmonic instruments. However, most of these models are rather general. These models typically assume that the musical signal is contaminated with white noise whose energy is statistically independent of the signal. See for instance [29] and the references therein. In contrast, we assume that there is a leading instrument, or soloist, that is accompanied by an orchestra or a chorus. The energy of the accompaniment is typically highly correlated with the energy of the soloist. Put another way, the dynamics of the accompaniment matches the dynamics of the soloist. For instance, when the soloist sings pianissimo the chorus follows her with pianissimo voices. We therefore developed a simple model whose parameters can be efficiently estimated that copes with the correlation in energy between the leading soloist and the accompaniment. In Fig. (4-2) we show the spectrum of one frame of a performance signal from our database. The harmonics of the soloist are designated by dashed lines. It is clear from the figure that there is a large concentration of energy at the designated harmonics. The residual energy, outside the harmonics, is certainly non-negligible but is clearly lower than the energy of the harmonics. Thus, our assumptions, although simplistic, seem to capture to a large extent the characteristics of the spectrum of singing with accompaniment.

Using the definition of a block $\bar{o}_i$ from chapter (3), the likelihood of the signal given the sequences of pitches and durations can be decomposed into a product of likelihood values of the individual blocks,

$$P(\boldsymbol{o}|\widetilde{\boldsymbol{d}}, \boldsymbol{p}) = \prod_{i=1}^{k} P(\overline{o}_i|p_i) \quad .$$

Therefore, the core of our modeling approach is a probabilistic model for the spectral distribution of a whole block given the underlying pitch frequency of the soloist. Our starting point is similar to the model presented in [29]. We assume that a note with pitch p_i attains high energy at frequencies which are multiples of p_i, namely at $p_i h$ for integer h. These frequencies are often referred to as harmonics. We explained the logic behind this assumption in Sec. (2.1.8). Since our signal is band limited, we only need to consider a finite set of harmonics h, $h \in \{1, 2, ..., H\}$. For practical purposes we set H to be 20 which enables a fast parameter estimation procedure. Let $F(\omega)$ denote the observed energy of the block $\overline{o}_i$ at frequency ω. Let $S(\omega)$ denote the energy of the soloist at frequency ω. The harmonic model assumes that $S(\cdot)$ is a scaled pulse train (see Sec. (2.1.8)), i.e., $S(\cdot)$ is bursts of energy centered at the harmonics of the pitch frequency, $p_i h$, and we model it as a weighted sum of delta functions,

$$S(\omega) = \sum_{h=1}^{H} A(h)\delta(p_i h - \omega) \quad , \tag{4.2}$$

where $A(h)$ is the volume gain for the harmonic whose index is h. The residual of the spectrum at frequency ω is denoted $N(\omega)$ and is equal to $N(\omega) = F(\omega) - S(\omega)$. We now describe a probabilistic model that leads to the following log-likelihood score,

$$\log P(\overline{o}_i|p_i) \propto \log \frac{\|S\|^2}{\|N\|^2} \quad ,$$

where $\|\cdot\|$ denotes the ℓ_2-norm.

To derive the above equation we assume that the spectrum of the ith block, F, is comprised of two components. The first component is the energy of the soloist, $S(\omega)$ as defined in Equ. (4.2). The second component is a general masking noise that encompasses the signal's energy due to the accompaniment and affects the entire spectrum. We denote the noise energy at frequency ω as $\eta(\omega)$. The energy of the spectrum at frequency ω is therefore modeled as,

$$F(\omega) = \sum_{h=1}^{H} A(h) \left(\eta(\omega) + \delta(p_i h - \omega)\right) \quad . \tag{4.3}$$

We now impose another simplifying assumption by setting the noise η to be a mul-

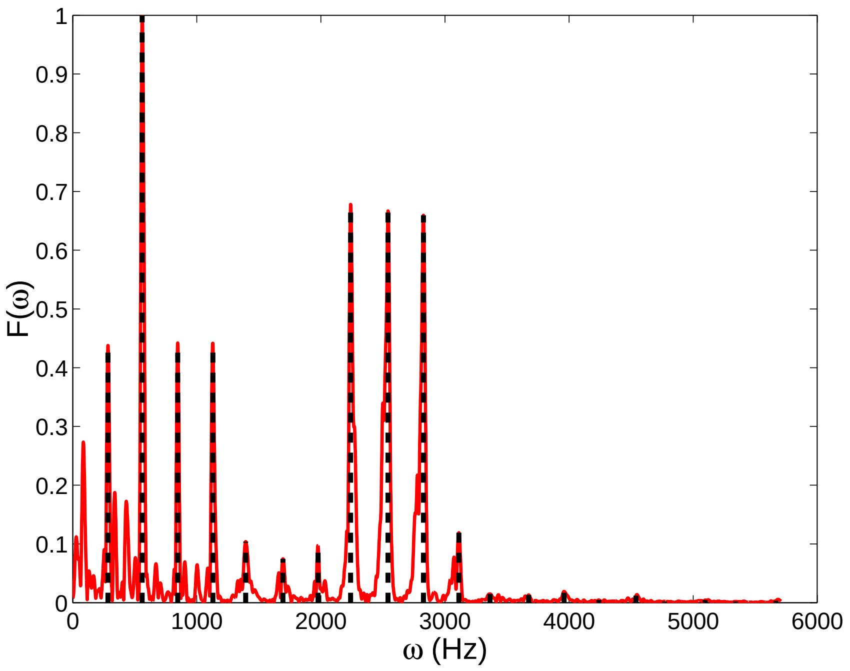

Figure 4-2: The spectrum of a single frame along with an impulse train designating the harmonics of the soloist.

tivariate normal random variable and further assuming that the noise values at each frequency ω are statistically independent with equal variance. Thus, the noise density function is

$$f(\eta|v) = \frac{1}{(2\pi v)^{L/2}} e^{-\frac{\|\eta\|^2}{2v}}$$

where v is the variance and L is the number of spectral points computed by the discrete Fourier transform. (We chose $L = 2^{15}$ to get a good spectral resolution.) Taking the log of the above density function we get,

$$\log f(\eta|v) = -\frac{L}{2}\log(2\pi v) - \frac{\|\eta\|^2}{2v} \quad . \tag{4.4}$$

The gain values $A(h)$ are free parameters which we need to estimate from the spectrum. Assuming that the noise level is relatively small compared to the bursts of energy at the harmonics of the pitch frequency, we set the value of $A(h)$ to be $F(p_i h)$. We also do not know the noise variance v. For this free parameter we use the simple maximum likelihood (ML) estimate which can be easily found as follows. The maximum likelihood estimate of v is found by taking the derivative of $\log f(\eta|v)$ with respect to v,

$$\frac{\partial \log f(\eta|v)}{\partial v} = -\frac{L}{2}\frac{2\pi}{2\pi v} + \frac{\|\eta\|^2}{2v^2} = 0 \quad \Rightarrow v^* = \frac{\|\eta\|^2}{L} \quad .$$

Rearranging Equ. (4.3), the noise value at frequency ω, $\eta(\omega)$, can be written as,

$$\eta(\omega) = \frac{F(\omega) - \sum_{h=1}^{H} A(h)\delta(p_i h - \omega)}{\sum_{h=1}^{H} A(h)} \quad .$$

By using above equation for $\eta(\omega)$ along with values set for $A(h)$ and the maximum likelihood estimate v^* in Equ. (4.4) we get,

$$\begin{aligned} \log f(\eta|v*) &= -\frac{L}{2}(\log(2\pi) + \log(\|\eta\|^2) - \log(L) - 1) \\ &= c + \frac{L}{2}\log\left(\frac{\|S\|^2}{\|N\|^2}\right) \quad . \end{aligned}$$

Since the stochastic ingredient of our spectral model is the accompanying noise, the noise likelihood above also constitute the likelihood of the spectrum,

$$\log P(\overline{o}_i|p_i) \propto \log\left(\frac{\|S\|^2}{\|N\|^2}\right) \quad . \qquad (4.5)$$

Calculating the probability using the STFT

We now explain how to calculate the probability in Equ. (4.5). The straight-forward implementation is as follows: given a signal portion $\overline{o}_i$ and a pitch value p_i, we calculate the spectrum of $\overline{o}_i$ using the DFT. However, because we analyse only a finite portion of the signal, we in fact calculate the spectrum of a multiplication of the signal with a rectangular window. That is, we get the convolution of the spectrum of the signal with the spectrum of a rectangular window (see Sec. (2.1.7)). Therefore, we need to consider the width of the main lobe and the leakage of the spectrum of the window function. In order to decrease the leakage effect, we can multiply the signal with smoother window functions (in our experiments we used *hanning* window [24, 16]). In order to overcome the problem of the main lobe, we need to sum over the main lobe width around each frequency of harmony, in the calculations of S and N. The properties of the spectrum of the window depend on the window length, so we need to calculate these properties according to the length of the relevant note.

Another option is first to calculate the STFT for fixed window lengths (see Sec. (2.1.7)). Then, we can sum the spectrum over the relevant windows in the STFT for obtaining the spectrum contents for a signal portion.

Chapter 5

Alignment and Retrieval

Let us overview our approach for retrieval. We are given a melody $(\boldsymbol{p}, \boldsymbol{d})$ and we want to find an audio signal $\boldsymbol{o}$ which represents a performance of this melody. Using our probabilistic framework, we cast the problem as the problem of finding a signal portion $\boldsymbol{o}$ whose likelihood given the melody, $P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d})$, is high. Our search strategy is as follows. We find the best alignment of the signal to the melody as we describe in the following section. The devised score of the alignment procedure is our means for retrieval. We then rank the segments of signals in accordance with their likelihood scores and return the segments achieving high likelihood scores.

5.1 Alignment

Alignment of a melody to a signal is performed by finding the best assignment of a tempo sequence. Formally, we are looking for the scaling factors $\boldsymbol{m}^*$ that attain the highest likelihood score, $\boldsymbol{m}^* = \arg\max_{\boldsymbol{m}} P(\boldsymbol{o}, \boldsymbol{m}|\boldsymbol{d}, \boldsymbol{p})$. Although the number of possible sequences of scaling factors $\boldsymbol{m}$ grows exponentially with the sequence length, the problem of finding $\boldsymbol{m}^*$ can be efficiently solved using dynamic programming, as we now describe.

Let $\boldsymbol{m}^i = (m_1, ..., m_i)$ denote the scaling factors of the first i notes of a melody. Let $\boldsymbol{o}^t = (o_1, ..., o_t)$ denote the first t samples of a signal. Let M be a discrete set of possible scaling factor values. For $\xi \in M$, let $M_{i,t,\xi}$ be a set of all possible sequences of i scaling factors, $\boldsymbol{m}^i$, such that $m_i = \xi$ is the scaling factor of note i and $t = \sum_{j=1}^{i} m_j d_j$ is the actual ending time of note i. Let $\gamma(i, t, \xi)$ be the joint likelihood of $\boldsymbol{o}^t$ and $\boldsymbol{m}^i \in M_{i,t,\xi}$

$$\gamma(i, t, \xi) = \max_{\boldsymbol{m}^i \in M_{i,t,\xi}} P(\boldsymbol{o}^t, \boldsymbol{m}^i|\boldsymbol{p}, \boldsymbol{d})$$

1. **Initialization**

$$\forall_{1\leq t\leq T}, \gamma(0, t, 1) = 1$$

2. **Recursion**

$$\gamma(i, t, \xi) = \max_{\xi' \in M} \gamma(i-1, t', \xi') P(\xi|\xi') P(o_{t'+1}, \ldots, o_t | p_i)$$

where $t' = t - d_i\xi$.

3. **Termination**

$$P^* = \max_{1\leq t\leq T,\ \xi\in M} \gamma(k, t, \xi)$$

Figure 5-1: The alignment algorithm.

The pseudo code for computing $\gamma(i, t, \xi)$ recursively is shown in Fig. (5-1).
The most likely sequence of scaling factors $\boldsymbol{m}^*$ is obtained from the algorithm by saving the intermediate values that maximize each expression in the recursion step. The complexity of the algorithm is $O(kT|M|^2D)$, where k is the number of notes, T is the number of samples in the digital signal, $|M|$ is the number of all possible tempo values and D is the maximal duration of a note. Using a pre-computation of the likelihood values we can reduce the time complexity by a factor of D and thus the run time of the algorithm reduces to $O(kT|M|^2)$. It is important to clarify that the pre-computation does not completely determine a single pitch value for a frame. It calculates the probability of the frame given each possible pitch in the diapason.

5.2 Retrieval

As mentioned above, our primary goal is to retrieve the segments of signals representing the melody given by the query. Theoretically, we need to assign a segment $\boldsymbol{o}$ its likelihood score, $P(\boldsymbol{o}|\boldsymbol{p}, \boldsymbol{d}) = \sum_{\boldsymbol{m}} P(\boldsymbol{m}|\boldsymbol{p}, \boldsymbol{d})$. However, this marginal probability is rather expensive to compute. We thus approximate this probability with the joint probability of the signal and most likely sequence of scaling factors, $P(\boldsymbol{o}, \boldsymbol{m}^*|\boldsymbol{p}, \boldsymbol{d})$. That is, we use the likelihood score of the alignment procedure as a retrieval score.

Chapter 6

Experimental Results

To evaluate our algorithm we collected 50 different melodies from famous opera arias, and queried these melodies in a database of real recordings. The recordings consist of 832 performances of opera arias performed by more than 40 different tenor singers with full orchestral accompaniment. Each performance is one minute. The data was extracted from seven audio CDs [3, 1, 5, 2, 4], and saved in *wav* format. Most of the performances (about 90 percent) are digital recordings (DDD/ADD). Yet, some performances are digital remastering of old analog recordings (AAD). This introduced additional complexity to the retrieval task due to varying level of noise.

The melodies for the experiments were extracted from MIDI files. About half of the MIDI files were downloaded from the Internet [1] and the rest of the MIDI files were performed on a MIDI keyboard and saved as MIDI files.

We compared three different tempo-based approaches for retrieval. The first method simply uses the original durations given in the query without any scaling. We refer to this simplistic approach as the *Fixed Tempo* (FT) model. The second approach uses a single scaling factor for all the durations of a given melody. However, this scaling factor is determined independently for each signal so as to maximize the signals likelihood. We refer to this model as the *Locally Fixed Tempo* (LFT) model. The third retrieval method is our variable tempo model that we introduced in this paper. We therefore refer to this method as the *Variable Tempo* (VT) model. By taking a prefix subset of each melody used in a query we evaluated three different lengths of melodies: 5 seconds, 15 seconds, and 25 seconds.

To assess the quality of the spectral distribution model described in Sec. (4.2), we implemented the spectral distribution model described in [29]. We describe this

[1] http://www.aria-database.com,
http://www.musicscore.freeserve.co.uk,
http://www.classicalmidi.gothere.uk.com

			Spectral Distribution Model					
			HSN			HIN		
			AvgP	Cov	Oerr	AvgP	Cov	Oerr
Melody length (sec.)	25	VT	**0.95**	**0.21**	**0.08**	0.92	0.40	0.10
		LFT	0.66	5.90	0.46	0.63	5.98	0.48
		FT	0.34	20.69	0.77	0.33	22.46	0.79
	15	VT	**0.86**	**1.75**	**0.19**	0.83	3.02	0.19
		LFT	0.66	8.10	0.44	0.66	8.15	0.42
		FT	0.38	19.83	0.71	0.36	19.08	0.73
	5	VT	**0.51**	**10.67**	**0.65**	0.46	11.83	0.69
		LFT	0.43	17.33	0.69	0.37	17.94	0.75
		FT	0.38	22.96	0.69	0.35	21.67	0.75

Table 6.1: Retrieval results

model in appendix (A). This model assumes that the harmonics of the signal are contaminated with noise whose mean energy is independent of the energy of the harmonics. We refer to our model as the *Harmonics with Scaled Noise* (HSN) model and to the model from [29] as the *Harmonics with Independent Noise* (HIN) model.

To evaluate the performances of the methods we used three evaluation measures: *one-error*, *coverage* and *average precision*. To explain these measures we introduce the following notation. Let N be the number of performances in our database and let M be the number of melodies that we search for. (As mentioned above, in our experiments $N = 832$ and $M = 50$.) For a melody index i we denote by Y_i the set of the performances containing melody i. The probabilistic modeling we discussed in this paper induces a natural ordering over the performances for each melody. Let $R_i(j)$ denote the ranking of the performance indexed j with respect to melody i. Based on the above definitions we now give the formal definitions of the performance measures we used for evaluation.

One-Error. The one-error measures how many times the top-ranked performance did *not* contain the melody posed in the query. Thus, if the goal of our system is to return a single performance that contains the melody, the one-error measures how many times the retrieved performance did not contain the melody. Formally, the definition of the one-error is,

$$\text{Oerr} = \frac{1}{M}\sum_{i=1}^{M}[\![\arg\min_j R_i(j) \notin Y_i]\!] \ .$$

where $[\![\pi]\!] = 1$ if predicate π holds and 0 otherwise.

Coverage. While the one-error evaluates the performance of a system with respect to the top-ranked performance, the goal of the coverage measure is to assess the performance of the system for all of the possible performances of a melody. Informally, Coverage measures the number of excess (non-relevant) performances we need to scan until we retrieve all the relevant performances. Formally, Coverage is defined as,

$$\mathrm{Cov} = \frac{1}{M}\sum_{i=1}^{M}(\max_{j\in Y_i} R_i(j) - |Y_i|) \ .$$

Average Precision. The above measures do not suffice in evaluating the performances of retrieval systems as one can achieve good (low) coverage but suffer high one-error rates, and vice versa. In order to assess the ranking performance as a whole we use the frequently used average precision measure. Formally, the average precision is defined as,

$$\mathrm{AvgP} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|Y_i|}\sum_{j\in Y_i}\frac{|\{j' \in Y_i | R_i(j') \leq R_i(j)\}|}{R_i(j)} \ .$$

In addition we also use precision versus recall graphs to illustrate the overall performances of the different approaches discussed in the paper. A precision-recall graph shows the level of precision for different recall values. The precision at k is the number of hits in the first k positions in the ranked list divided by k. Formally,

$$Precision(k) = \frac{1}{M}\sum_{i=1}^{M}\frac{|\{R_i(j) \ : \ (j \in Y_i) \wedge (R_i(j) \leq k)\}|}{k} \ .$$

The recall at k is the number of hits in the first k positions in the ranked list divided by the number of all the relevant performances in the database. Formally,

$$Recall(k) = \frac{1}{M}\sum_{i=1}^{M}\frac{|\{R_i(j) \ : \ (j \in Y_i) \wedge (R_i(j) \leq k)\}|}{|Y_i|} \ .$$

The graphs presented in this paper are non-interpolated, that is, they were calculated based on the precision and recall values achieved at integer positions of the ranked lists.

In table 6.1 we report results with respect to the performance measures described for the FT, LFT, and VT models. For each tempo model we conducted the experiments with the two spectral distribution models HIN and HSN. It is clear from the table that the *Variable Tempo* model with the *Harmonics with Scaled Noise* spectral

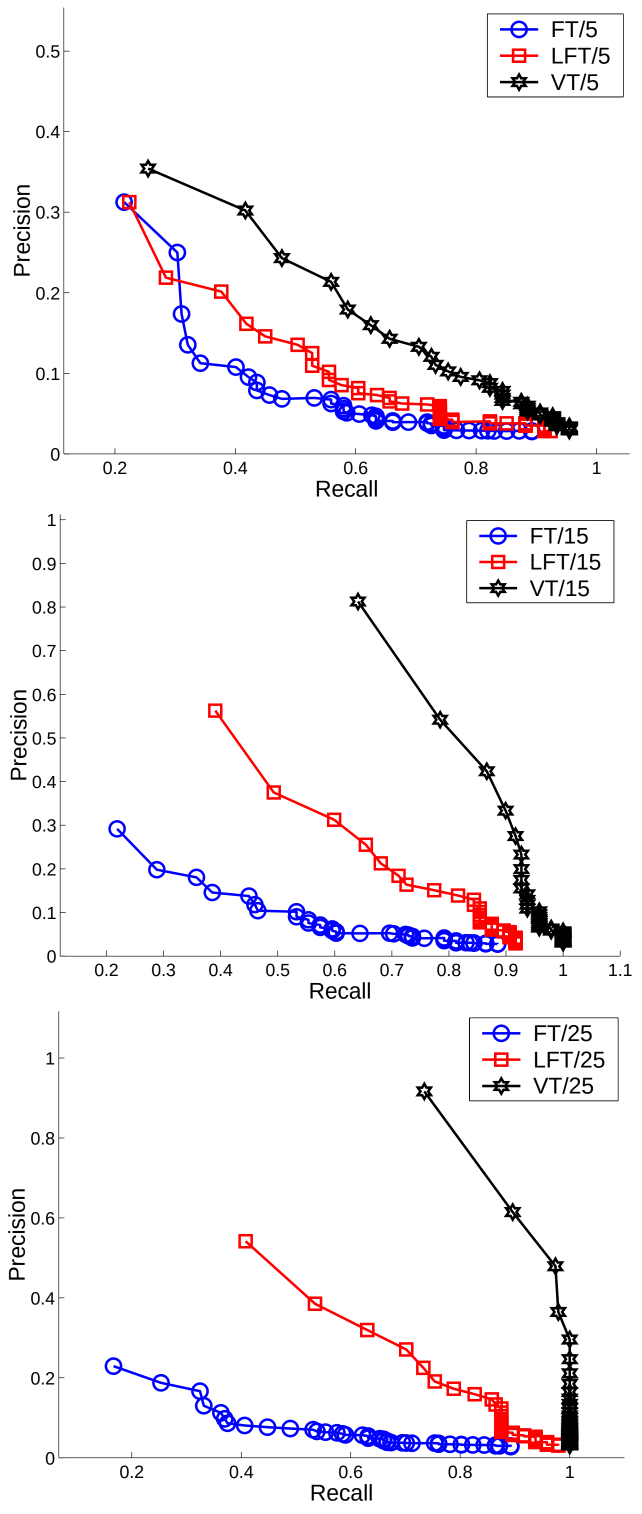

Figure 6-1: Precision-recall curves comparing the performance of three tempo models for queries consisting of five seconds (top), fifteen second (middle), and twenty five seconds (bottom).

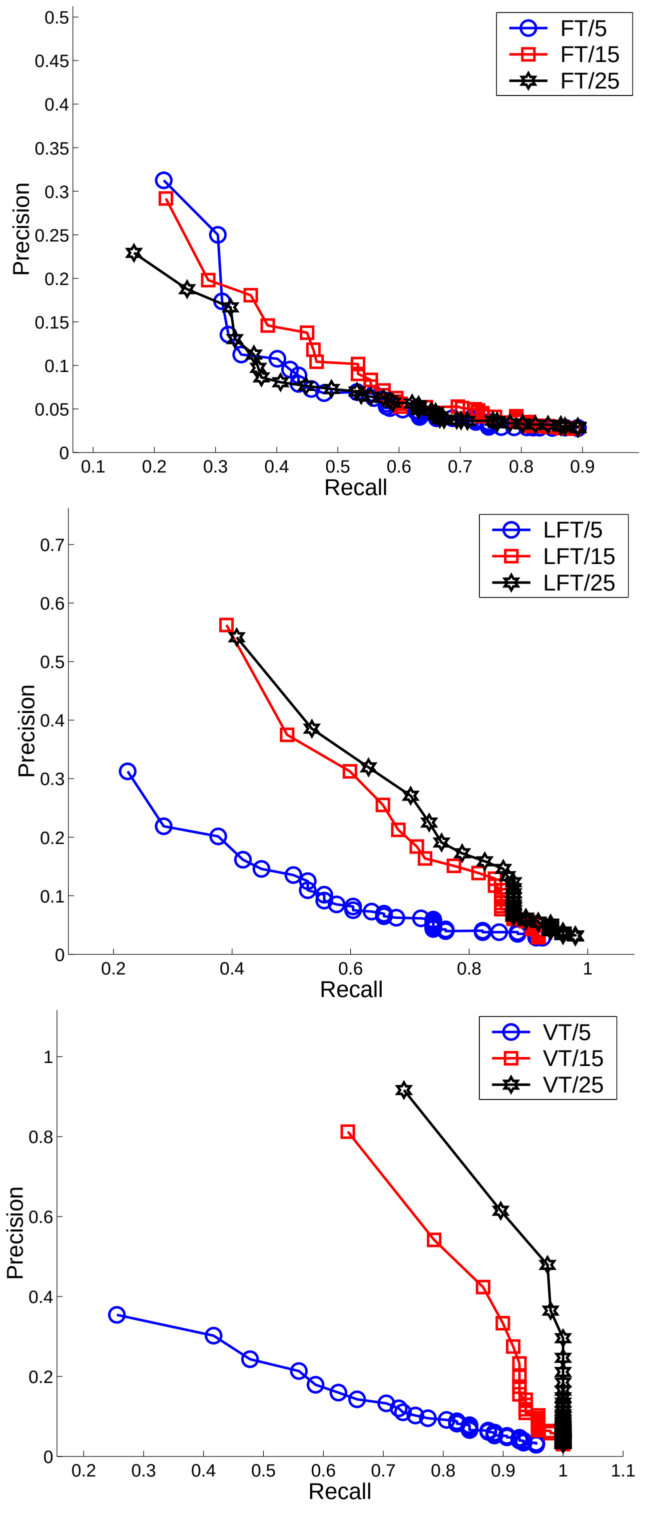

Figure 6-2: Precision-recall curves comparing the performance of each of the tempo models for three different query lengths.

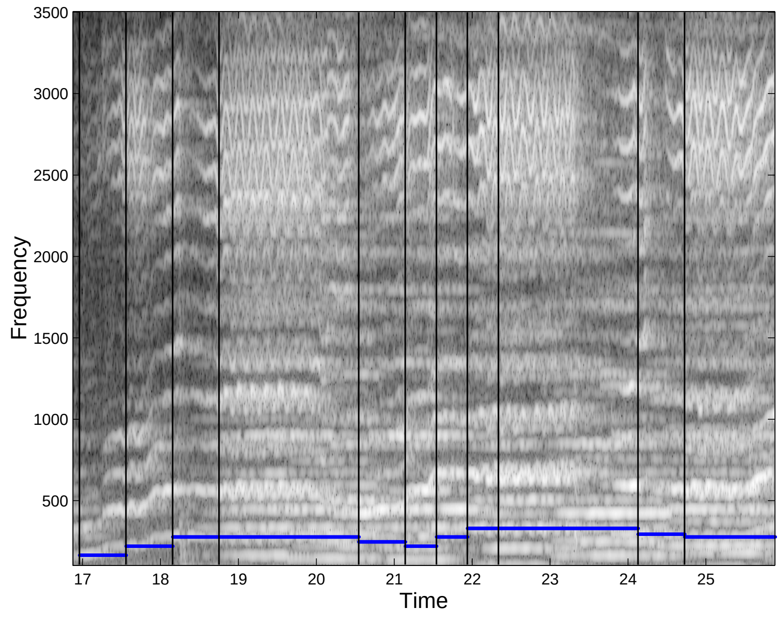

Figure 6-3: An illustration of the alignment and segmentation of the VT model. The pitches of the notes in the melody are overlayed in solid lines.

distribution outperforms the rest of the models and achieves superior results. Moreover, the performance of the *Variable Tempo* model consistently improves as the duration of the queries increases. In contrast, the *Fixed Tempo* does not exhibit any improvement as the duration of the queries increases and the *Locally Fixed Tempo* shows only a moderate improvement when using fifteen second long queries instead of five second long queries and it does not improve as the duration grows to twenty five seconds. A reasonable explanation for these phenomena is that the amount of variability in a very short query is naturally limited and thus the leverage gained by accurate tempo modeling which takes into account the variability in tempo is rather small. Thus, as the query duration grows the power of the variable tempo model is better exploited. The *Locally Fixed Tempo* can capture the average tempo of a performance but clearly fails to capture changes in the tempo. Since the chance of a tempo change grows with the duration of the query the average tempo stops from being a good approximation and we do not see further improvement in the retrieval quality.

In Fig. (6-1) we give precision-recall graphs that compare the three tempo models. Each graph compares FT, LFT and VT for different query durations. The VT model clearly outperforms both the FT and LFT models. The longer the query the wider the gap in performance. In Fig. (6-2) we compare the precision-recall graphs for each model as a function of the query duration. Each graph shows the precision-recall curves for 5, 15, and 25 seconds queries. We again see that only the VT model consistently improves with the increase in the query duration. Using a globally fixed tempo (FT) is clearly inadequate as it results in very poor performance – precision is never higher than 0.35 even for low level of recall. The performance of the LFT model is more reasonable. A precision of about 0.5 can be achieved for a recall value of 0.5. However, the full power of our approach is utilized only when we use the VT model. We achieve an average precision of 0.92 with a recall of 0.75. It seems that with the VT model we reach an overall performance that can serve as the basis for large scale music retrieval systems.

Lastly, as a final sanity check of the conjecture of the robustness of the VT model we used the VT and LFT model with three long melody queries (one minute) and applied the retrieval and alignment process. We then let a professional musician listen to the segmentation and browse the segmented spectogram. An example of a spectogram with a segmentation of the VT model is given in Fig. (6-3). The example is of a performance where the energy of accompaniment is higher than the energy of the leading tenor. Nonetheless, a listening experiment verified that our system was able to properly segment and align the melody posed by the query. Although these perceptual listening tests are subjective, the experiments indicated that the VT model also provides an accurate alignment and segmentation.

Chapter 7

Discussion

In this work we presented a robust probabilistic model for query by melody. The proposed approach is simple to implement and was found to work well on polyphony-rich recordings with various types of accompaniments. The probabilistic model that we developed focuses on two main sources of variability. The first is variations in the actual durations of notes in real recordings (tempo variability) and the second is the variability of the spectrum mainly due to the "spectral masking" of the leading vocal by the accompanying vocals and orchestra. In this work we assumed that the pitch information in a query is accurate and only the duration can be altered in the performance. This assumption is reasonable if the queries are posed using a symbolic input mechanism such as a MIDI keyboard. However, an easier and more convenient mechanism is to hum or whistle a melody. This task is often called "query by humming". In addition to the tempo variability and spectral masking, a query by humming system also needs to take into account imperfections in the pitch of the hummed melody. Indeed, much of the work on query by humming have been devoted to music retrieval using noisy pitch information. The majority of the work on query by humming though, have focused on search of noisy queries in symbolic databases. Since the main thrust of this research is searches in real polyphonic recordings, it complements the research on query by humming and can supplement numerous systems that search in symbolic databases. We plan to extend our algorithm so it can be combined with a front end for hummed queries. In addition, we have started conducting research on supervised methods for musical genre classification. We believe that by combining highly accurate genre classification with a robust retrieval and alignment we will be able to provide an effective tool for searching and browsing for both professionals and amateurs.

Appendix A

The Harmonic with Independent Noise model

In this appendix we describe the Harmonic with Independent Noise (HIN) model. The original model was written in the time domain. However, since we introduced our model in the frequency domain, we also introduce the HIN model in the frequency domain. For the original presentation in the time domain see [29, 23, 11].

Let $F(\omega)$ be the spectrum of a note with pitch p. Using the harmonic model, we assume that the spectrum can be written as,

$$F(\omega) = \left(\sum_{h=1}^{H} A(h)\, \delta(ph - \omega) \right) + \eta(\omega) \quad , \tag{A.1}$$

where H, $\eta(\cdot)$ and $A(\cdot)$ are the same as in Sec. (4.2). Rearranging Equ. (A.1) we get the expression for the noise,

$$\eta(\omega) = F(\omega) - \left(\sum_{h=1}^{H} A(h)\, \delta(ph - \omega) \right) \quad . \tag{A.2}$$

We impose a simplifying assumption by setting the noise η to be a multivariate normal random variable and further assuming that the noise values at each frequency ω are statistically independent with equal variance. Thus, the noise density function is

$$f(\eta) = \frac{1}{(2\pi v)^{L/2}} e^{-\frac{\|\eta\|^2}{2v}}$$

where v is the variance and L is the number of spectral points computed by the discrete Fourier transform. Taking the log of the above density function we get,

$$\log f(\eta|v) = -\frac{L}{2}\log(2\pi v) - \frac{\|\eta\|^2}{2v} \quad . \tag{A.3}$$

We found the maximum likelihood estimation for the gain values $A(h)$ by taking the derivative of $\log f(\eta)$ with respect to $A(h)$,

$$\frac{\partial \log f(\eta)}{\partial A(h)} = 0 \quad \Rightarrow A^*(h) = F(ph) \quad .$$

Similarly, we found the maximum likelihood estimation for the variance v by taking the derivative of $\log f(\eta)$ with respect to v,

$$\frac{\partial \log f(\eta|v)}{\partial v} = 0 \quad \Rightarrow v^* = \frac{\|\eta\|^2}{L} \quad .$$

Plugging the above maximum likelihood estimators for A and v into Equ. (A.3) and using the expression for η in Equ. (A.2) we get

$$\begin{aligned} \log f(\eta|v*, A^*) &= -\frac{L}{2}(\log(2\pi) + \log(\|\eta\|^2) - \log(L) - 1) \\ &= c + \frac{L}{2}\log\left(\frac{1}{\|N\|^2}\right) \quad , \end{aligned} \tag{A.6}$$

where $N(\omega)$ is defined in Sec. (4.2).

Bibliography

[1] les 40 tenors. EMI 5720072 (Two CDs).

[2] Arien. famous tenor arias. Decca 450 005-2.

[3] Best of opera. Disky 701812.

[4] Luciano pavarotti. nessun dorma, arias and duets. Decca 467 462-2.

[5] The young domingo. BMG 63527 (Two CDs).

[6] J. Aucouturier and Sandler M. Segmentation of musical signals using hidden markov models. *Audio Engineering Society*, May 2001.

[7] D. Butler. *The musician's guide to perception and cognition.* Schirmer Books, 1992.

[8] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[9] R. Dannenberg. An on-line algorithm for real-time accompaniment. *Proc. Int'l Computer Music Conference*, 1984.

[10] T. Dean and K. Kanazawa. A model for reasoning about persistent and causation. *Computational Intelligence*, 5(3):142–150, 1989.

[11] M. Deisher and A. Spanias. Hmm-based speech enhancement using harmonic modeling. *ICASSP'97, Vol. 2, p. 1175*, 1997.

[12] A. S. Durey and M. A. Clements. Melody spotting using hidden markov models. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 109–117, Bloomington, IN, October 2001.

[13] J.M. Eargle. *Music, Sound, and Technology.* Van Nostrand Reinhold, 1990.

[14] J. Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999.

[15] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith. Query by humming: Musical information retrieval in an audio database. In *ACM Multimedia*, pages 231–236, 1995.

[16] V.K. Ingle and J.G. Proakis. *Digital Signal Processing using Matlab.* Brooks/Cole, 2000.

[17] D. Koller and N. Friedman. Introduction to probabilistic graphical models, handouts. http://www.cs.huji.ac.il/˜pmai.

[18] K. Lemstrom and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proc. Content-Based Multimedia Information Access (RIAO'2000)*, pages 1261–1279, April 2000.

[19] R. McNab, L. Smith, D. Bainbridge, and I. Witten. The new zealand digital library melody index, 1997.

[20] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Sunningham. Toward the digital music library: tune retrieval from acoustic input. In *Proc. ACM Digital Libraries*, 1996.

[21] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities, 24:161– 175*, 1990.

[22] A.V. Oppenheim and R.W. Schafer. *Digital Signal Processing.* Prentice-Hall, 1975.

[23] L. Parra and U. Jain. Approximate kalman filtering for the harmonic plus noise model. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[24] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing.* Prentice-Hall, 1996.

[25] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, January 1986.

[26] C. Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 21(4), April 1999.

[27] T. Robinson. Speech analysis, online tutorial.
http://svr-www.eng.cam.ac.uk/~ajr/SA95/.

[28] J.O. Smith. Mus320: Introduction to digital audio signal processing.
http://www-ccrma.stanford.edu/courses/320/Welcome.html.

[29] J. Tabrikian, S. Dubnov, and Y. Dickalov. Speech enhancement by harmonic modeling via map pitch tracking. *ICASSP2002, Orlando, Florida*, 2002.

[30] G. Tzanetakis and P. Cook. Audio information retrieval (air) tools. In *Proc. International Symposium on Music Information Retrieval*, 2000.

[31] P.J. Walmsley, S.J. Godsill, and P.J.W. Rayner. Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters. In *Proc. 1999 Ieee Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1999.