11

Music Scene Description

Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST).
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan. m.goto@aist.go.jp

11.1 Introduction

This chapter introduces a research approach called *'music scene description'* [232], [225], [228], where the goal is to build a computer system that can understand musical audio signals at the level of untrained human listeners without trying to extract every musical note from music. People listening to music can easily hum the melody, clap hands in time to the musical beat, notice a phrase being repeated, and find chorus sections. The brain mechanisms underlying these abilities, however, are not yet well understood. In addition, it has been difficult to implement these abilities on a computer system, although a system with them is useful in various applications such as music information retrieval, music production/editing, and music interfaces. It is therefore an important challenge to build a music scene description system that can understand complex real-world music signals like those recorded on commercially distributed compact discs (CDs).

Music scene description differs from two popular approaches to deal with music signals, sound source separation and traditional automatic music transcription (in the narrow sense[1]). Although these technologies are valuable from an engineering viewpoint, neither separation nor transcription is necessary or sufficient for understanding music.

- *It is possible to understand music without sound source separation.*
  The fact that human listeners understand various properties of audio signals is not necessarily evidence that the human auditory system extracts the audio signal of each individual source. Even if a mixture of two components cannot be separated, it can be understood from their salient features that the mixture includes them. In fact, from the viewpoint of auditory

[1]The term 'automatic music transcription' in this chapter refers to a traditional approach of transcribing all musical notes as a score, while the term 'automatic music transcription' in this book has a broader meaning including the music scene description as described in Chapter 1 of this volume.

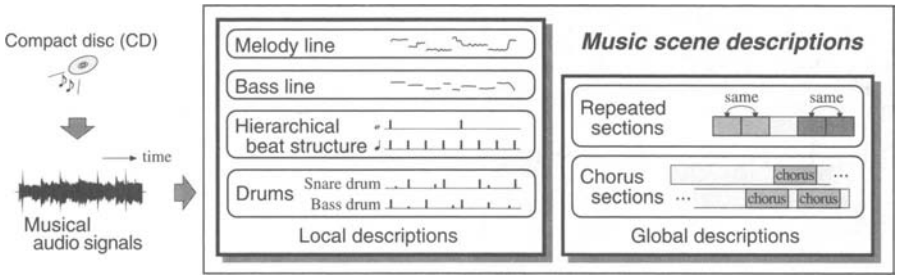


Fig. 11.1. Music scene descriptions.

psychology, it has been pointed out that human listeners do not perform sound source separation: perceptual sound source segregation is different from signal-level separation. For example, Bregman noted that 'there is evidence that the human brain does not completely separate sounds' [50]. The approach of developing methods for monaural or binaural sound source separation might deal with a hard problem which is not solved by any mechanism in this world (not solved even by the human brain).

- *It is possible to understand music without complete music transcription.* Music transcription, identifying the names (symbols) of musical notes, is a difficult skill mastered only by trained musicians. As pointed out by Goto [239], [240], [232] and Scheirer [565], untrained listeners understand music to some extent without mentally representing audio signals as musical scores. For example, as known from the observation that a listener who cannot identify the name and constituent notes of a chord can nevertheless feel the harmony and chord changes, a chord is perceived as combined whole sounds (tone colour) without reducing it to its constituent notes (like reductionism). Furthermore, even if it is possible to derive separated signals and musical notes, it would still be difficult to obtain high-level music descriptions like melody lines and chorus sections.

The music scene description approach therefore emphasizes methods that can obtain a certain description of a music scene from sound mixtures of various musical instruments in a musical piece. Here, it is important to discuss what constitutes an appropriate *description* of music signals. Since various levels of abstraction for the description are possible, it is necessary to consider which level is an appropriate first step towards the ultimate description in human brains. Goto [232], [228] proposed the following three viewpoints:

- An intuitive description that can be easily obtained by untrained listeners.
- A basic description that trained musicians can use as a basis for higher-level music understanding.
- A useful description facilitating the development of various practical applications.

According to these viewpoints, the following local and global descriptions (Fig. 11.1) have been proposed for Western music:

1. *Melody and bass lines*
   Melody and bass lines represent the temporal trajectory of the melody and bass. The melody is a series of single tones and is heard more distinctly than the rest. The bass is a series of single tones and is the lowest frequency part in polyphonic music. Note that a melody or bass line here is not represented as a series of musical notes; it is a continuous representation of fundamental frequency (F0, perceived as pitch) and power transitions. Only music with distinct melody and bass lines is dealt with for this description.
2. *Hierarchical beat structure*
   Hierarchical beat structure represents the fundamental temporal structure of music and comprises the quarter-note (beat) and measure levels—i.e., the positions of quarter-note beats and bar lines (corresponding to the metrical levels of 'beat' and 'bar' in Fig. 4.1, p. 106).
3. *Drums*
   Drums represent onset times of principal drum sounds, such as bass and snare drums. Their temporal patterns form drum patterns. Only music with drum sounds is dealt with for this description.
4. *Chorus sections and repeated sections*
   Chorus sections represent the most representative, uplifting, and prominent thematic sections in the structure of a musical piece (especially in popular music). Since chorus sections are usually repeated, they are represented as a list of the start and end points of every chorus section. Repeated sections represent the repetition of temporal regions with various lengths. Only music with distinct repeated choruses, such as popular music, is dealt with for the description of chorus sections, while any music can be dealt with for the description of repeated sections.

The idea behind these descriptions came from introspective observation of how untrained listeners listen to music. The following sections introduce methods for producing these descriptions from music signals such as CD recordings, which contain simultaneous sounds of various instruments (with or without drum sounds). In general, these methods deal with monaural audio signals because stereo signals on CDs can be easily converted to monaural signals by averaging the left and right channels. While methods depending on stereo information [24] can have better performance than methods dealing with monaural signals, such stereo-based methods cannot be applied to monaural signals. Methods assuming monaural signals, on the other hand, can be applied to stereo signals and be considered essential to music understanding since human listeners have no difficulty understanding the above descriptions even from monaural signals.

11.2 Estimating Melody and Bass Lines

The estimation of melody and bass lines is important because the melody forms the core of Western music and is very influential in the identity of a musical piece, while the bass is closely related to the tonality (see Chapter 1). These lines are fundamental to the perception of music by both musically trained and untrained listeners. They are also useful in various applications such as automatic music indexing for information retrieval (e.g., searching for a song by singing a melody), computer participation in live human performances, musical performance analysis of outstanding recorded performances, and automatic production of accompaniment tracks for karaoke using CDs.

It is difficult to estimate the fundamental frequency (F0) of melody and bass lines in monaural sound mixtures from CD recordings. Most previous F0 estimation methods cannot be applied to this estimation because they require that the input audio signal contain just a single-pitch sound with aperiodic noise or that the number of simultaneous sounds be known beforehand. The main reason F0 estimation in sound mixtures is difficult is that, in the time-frequency domain, the frequency components of one sound often overlap the frequency components of simultaneous sounds. In popular music, for example, part of the voice's harmonic structure is often overlapped by harmonics (overtone partials) of the keyboard instrument or guitar, by higher harmonics of the bass guitar, and by noisy inharmonic frequency components of the snare drum. A simple method for locally tracing a frequency component is therefore neither reliable nor stable. Moreover, F0 estimation methods relying on the existence of the F0s frequency component (the frequency component corresponding to the F0) not only cannot handle the *missing fundamental*, but are also unreliable when the F0s frequency component is smeared by the harmonics of simultaneous sounds.

F0 estimation of melody and bass lines in CD recordings was first achieved in 1999 by Goto [232], [222], [228]. Goto proposed a real-time method called *PreFEst* (Predominant-F0 Estimation method) which estimates the melody and bass lines in monaural sound mixtures. Unlike previous F0 estimation methods, PreFEst does not assume the number of sound sources, locally trace frequency components, or even rely on the existence of the F0s frequency component. PreFEst basically estimates the F0 of the most predominant harmonic structure—the most predominant F0 corresponding to the melody or bass line—within an intentionally limited frequency range of the input mixture. It simultaneously takes into consideration all possibilities for the F0 and treats the input mixture as if it contained all possible harmonic structures with different weights (amplitudes). To enable the application of statistical methods, the input frequency components are represented as a probability density function (pdf), called an *observed pdf*. The point is that the method regards the observed pdf as a weighted mixture of harmonic-structure tone

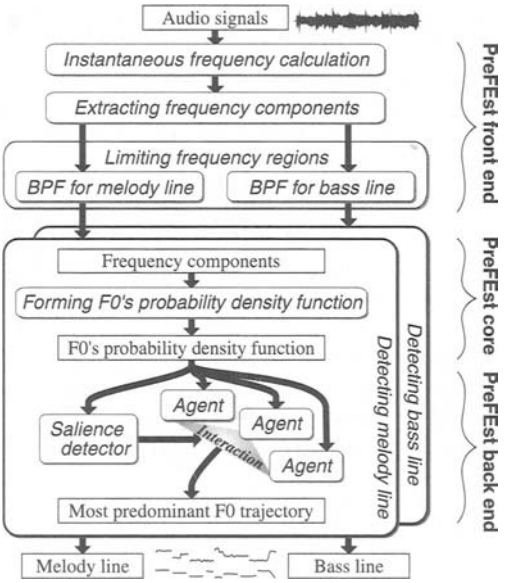


Fig. 11.2. Overview of PreFEst (Predominant-F0 Estimation method) for estimating melody and bass lines in CD recordings. In this figure, BPF denotes bandpass filtering.

models (represented by pdfs) of all possible F0s. It simultaneously estimates both their weights corresponding to the relative dominance of every possible harmonic structure and the shape of the tone models by maximum *a posteriori* probability (MAP) estimation (see Chapter 2, p. 40 for an introduction to MAP estimation methods) considering their prior distribution. It then considers the maximum-weight model as the most predominant harmonic structure and obtains its F0. The method also considers the F0s temporal continuity by using a multiple-agent architecture.

The following sections first explain the PreFEst method in detail and then introduce other methods for estimating the melody line developed by Paiva, Mendes, and Cardoso [494], [493], Marolt [435], [436], and Eggink and Brown [169], and a method for estimating the bass line developed by Hainsworth and Macleod [264]. Figure 11.2 shows an overview of PreFEst. PreFEst consists of three components, the *PreFEst front end* for frequency analysis, the *PreFEst core* to estimate the predominant F0, and the *PreFEst back end* to evaluate the temporal continuity of the F0. Since the melody line tends to have the most predominant harmonic structure in middle and high-frequency regions, and the bass line tends to have the most predominant harmonic structure in a low-frequency region, the F0s of the melody and bass lines can be estimated by applying the PreFEst core with appropriate frequency-range limitation.

11.2.1 PreFEst Front End: Forming the Observed Probability Density Functions

The PreFEst front end first uses a multirate filterbank to obtain adequate time-frequency resolution under a real-time constraint. By using an instantaneous frequency-related measure [84], [7], [338] for the existence of frequency components, it then extracts frequency components $\Psi^{(t)}(\nu)$ from the short-time Fourier transform (STFT) $X(\nu, t)$ of a signal

$$\Psi^{(t)}(\nu) = \begin{cases} |X(\nu,t)| & \text{if } \nu \text{ has a frequency component,} \\ 0 & \text{otherwise,} \end{cases} \tag{11.1}$$

where t is the time measured in units of frame-shifts (10 ms), and ν is the log-scale frequency denoted in units of *cents* (a musical-interval measurement). Frequency f_{Hz} in Hertz is converted to frequency f_{cent} in cents so that there are 100 cents to a tempered semitone and 1200 to an octave:

$$f_{\mathrm{cent}} = 1200 \log_2 \frac{f_{\mathrm{Hz}}}{440 \times 2^{\frac{3}{12}-5}}. \tag{11.2}$$

To obtain two sets of bandpass-filtered frequency components, one for the melody line (261.6–4186 Hz) and the other for the bass line (32.7–261.6 Hz) [228],[2] the PreFEst front end uses bandpass filters (BPFs) whose frequency response is $\mathsf{BPF}_u(\nu)$ where u denotes the melody line (u = 'melody') or the bass line (u = 'bass line'). Each set of the bandpass-filtered components is finally represented as an *observed pdf* $\mathsf{p}_{\Psi}^{(t)}(\nu)$

$$\mathsf{p}_{\Psi}^{(t)}(\nu) = \frac{\mathsf{BPF}_u(\nu)\, \Psi^{(t)}(\nu)}{\int_{-\infty}^{\infty} \mathsf{BPF}_u(\eta)\, \Psi^{(t)}(\eta)\, d\eta}. \tag{11.3}$$

11.2.2 PreFEst Core: Estimating the F0s Probability Density Function

For each melody or bass line set of filtered frequency components represented as an observed pdf $\mathsf{p}_{\Psi}^{(t)}(\nu)$, the PreFEst core forms a probability density function of the F0, called the *F0s pdf*, $\mathsf{p}_{F0}^{(t)}(\nu_0)$, where ν_0 is the log-scale fundamental frequency in cents. The PreFEst core considers each observed pdf to have been generated from a weighted-mixture model of the tone models of all possible F0s; the tone model is the pdf corresponding to a typical harmonic structure and indicates where the harmonics (overtone partials) of the F0 tend to occur (Fig. 11.3). Because the weights of tone models represent the relative dominance of every possible harmonic structure, these weights can be regarded as the F0s pdf: the more dominant a tone model is in the mixture, the higher the probability of the F0 of its model.

[2]The method finds the F0 whose harmonics are most predominant in those limited frequency ranges. In other words, whether the F0 is within each limited range or not, PreFEst tries to estimate the F0 which is supported by predominant harmonic frequency components within that range.

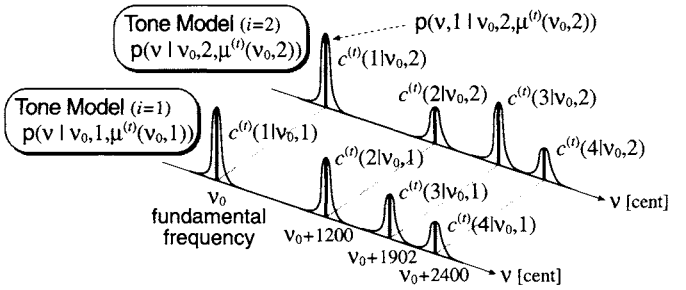


Fig. 11.3. Model parameters of multiple adaptive tone models $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))$.

Weighted-Mixture Model of Adaptive Tone Models

To deal with diversity of the harmonic structure, the PreFEst core can use several types of harmonic-structure tone models. The pdf of the i-th tone model for each F0 ν_0 is denoted by $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))$ (see Fig. 11.3), where the model parameter $\boldsymbol{\mu}^{(t)}(\nu_0, i)$ represents the shape of the tone model. The number of tone models is I_u (that is, $i = 1, \ldots, \mathrm{I}_u$), where u denotes the melody line (u = 'melody') or the bass line (u = 'bass line'). Each tone model is defined by

$$\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i)) = \sum_{m=1}^{M_u} \mathsf{p}(\nu, m|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i)), \tag{11.4}$$

$$\mathsf{p}(\nu, m|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i)) = c^{(t)}(m|\nu_0, i)\, \mathcal{N}\big(\nu; \nu_0 + 1200 \log_2 m, \sigma_u^2\big), \tag{11.5}$$

$$\boldsymbol{\mu}^{(t)}(\nu_0, i) = \{c^{(t)}(m|\nu_0, i) \mid m = 1, \ldots, M_u\}, \tag{11.6}$$

where M_u is the number of harmonics considered, σ_u^2 is the variance of the Gaussian distribution $\mathcal{N}\big(\nu; \nu_0, \sigma_u^2\big)$ (see (2.16), p. 29 for a definition), and $c^{(t)}(m|\nu_0, i)$ determines the relative amplitude of the m-th harmonic component (the shape of the tone model) and satisfies

$$\sum_{m=1}^{M_u} c^{(t)}(m|\nu_0, i) = 1. \tag{11.7}$$

In short, this tone model places a weighted Gaussian distribution at the position of each harmonic component.

The PreFEst core then considers the observed pdf $\mathsf{p}_\Psi^{(t)}(\nu)$ to have been generated from the following model $\mathsf{p}(\nu|\boldsymbol{\theta}^{(t)})$, which is a weighted mixture of all possible tone models $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))$:

$$\mathsf{p}(\nu|\boldsymbol{\theta}^{(t)}) = \int_{\mathrm{F}_u^{\mathrm{l}}}^{\mathrm{F}_u^{\mathrm{h}}} \sum_{i=1}^{\mathrm{I}_u} w^{(t)}(\nu_0, i)\, \mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))\, d\nu_0, \tag{11.8}$$

$$\boldsymbol{\theta}^{(t)} = \{\boldsymbol{w}^{(t)}, \boldsymbol{\mu}^{(t)}\}, \tag{11.9}$$

$$\boldsymbol{w}^{(t)} = \{w^{(t)}(\nu_0, i) \mid \mathrm{F}_u^{\mathrm{l}} \leq \nu_0 \leq \mathrm{F}_u^{\mathrm{h}}, i = 1, \ldots, \mathrm{I}_u\}, \tag{11.10}$$

$$\boldsymbol{\mu}^{(t)} = \{\boldsymbol{\mu}^{(t)}(\nu_0, i) \mid \mathrm{F}_u^{\mathrm{l}} \leq \nu_0 \leq \mathrm{F}_u^{\mathrm{h}}, i = 1, \ldots, \mathrm{I}_u\}, \tag{11.11}$$

where $\mathrm{F}_u^{\mathrm{l}}$ and $\mathrm{F}_u^{\mathrm{h}}$ denote the lower and upper limits of the possible (allowable) F0 range and $w^{(t)}(\nu_0, i)$ is the weight of a tone model $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))$ that satisfies

$$\int_{\mathrm{F}_u^{\mathrm{l}}}^{\mathrm{F}_u^{\mathrm{h}}} \sum_{i=1}^{\mathrm{I}_u} w^{(t)}(\nu_0, i)\, d\nu_0 = 1. \tag{11.12}$$

Because the number of sound sources cannot be known *a priori*, it is important to simultaneously take into consideration all F0 possibilities as expressed in (11.8). If it is possible to estimate the model parameter $\boldsymbol{\theta}^{(t)}$ such that the observed pdf $\mathsf{p}_{\Psi}^{(t)}(\nu)$ is likely to have been generated from the model $\mathsf{p}(\nu|\boldsymbol{\theta}^{(t)})$, the weight $w^{(t)}(\nu_0, i)$ can be interpreted as the F0s pdf $\mathsf{p}_{F0}^{(t)}(\nu_0)$:

$$\mathsf{p}_{F0}^{(t)}(\nu_0) = \sum_{i=1}^{\mathrm{I}_u} w^{(t)}(\nu_0, i) \qquad (\mathrm{F}_u^{\mathrm{l}} \leq \nu_0 \leq \mathrm{F}_u^{\mathrm{h}}). \tag{11.13}$$

Introducing a Prior Distribution

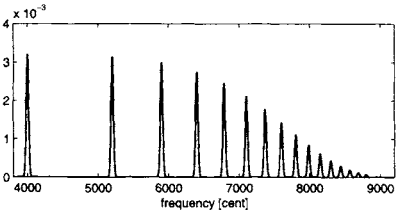
To use prior knowledge about F0 estimates and the tone model shapes, a prior distribution $\mathsf{p}_{0u}(\boldsymbol{\theta}^{(t)})$ of $\boldsymbol{\theta}^{(t)}$ is defined as follows:

$$\mathsf{p}_{0u}(\boldsymbol{\theta}^{(t)}) = \mathsf{p}_{0u}(\boldsymbol{w}^{(t)})\, \mathsf{p}_{0u}(\boldsymbol{\mu}^{(t)}), \tag{11.14}$$

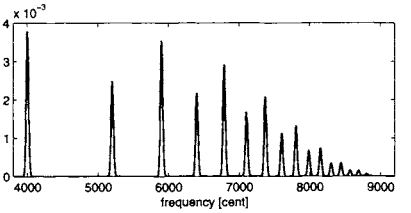
$$\mathsf{p}_{0u}(\boldsymbol{w}^{(t)}) = \frac{1}{Z_{\boldsymbol{w}}} e^{-\beta_{\boldsymbol{w}u}^{(t)}\, \mathsf{D}_{\boldsymbol{w}}(\boldsymbol{w}_{0u}^{(t)}; \boldsymbol{w}^{(t)})}, \tag{11.15}$$

$$\mathsf{p}_{0u}(\boldsymbol{\mu}^{(t)}) = \frac{1}{Z_{\boldsymbol{\mu}}} e^{-\int_{\mathrm{F}_u^{\mathrm{l}}}^{\mathrm{F}_u^{\mathrm{h}}} \sum_{i=1}^{\mathrm{I}_u} \beta_{\boldsymbol{\mu}u}^{(t)}(\nu_0, i)\, \mathsf{D}_{\boldsymbol{\mu}}(\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i); \boldsymbol{\mu}^{(t)}(\nu_0, i))\, d\nu_0}. \tag{11.16}$$

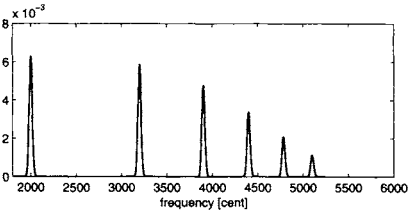
Here, $\mathsf{p}_{0u}(\boldsymbol{w}^{(t)})$ and $\mathsf{p}_{0u}(\boldsymbol{\mu}^{(t)})$ are unimodal distributions: $\mathsf{p}_{0u}(\boldsymbol{w}^{(t)})$ takes its maximum value at $w_{0u}^{(t)}(\nu_0, i)$ and $\mathsf{p}_{0u}(\boldsymbol{\mu}^{(t)})$ takes its maximum value at $\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i)$ $(= \{c_{0u}^{(t)}(m|\nu_0, i) \mid m = 1, \ldots, M_u\})$, where $w_{0u}^{(t)}(\nu_0, i)$ and $\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i)$ are the most probable parameters. Figure 11.4 shows two examples of the most probable tone model shape parameters, $\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i)$, used in Goto's implementation. $Z_{\boldsymbol{w}}$ and $Z_{\boldsymbol{\mu}}$ are normalization factors, and $\beta_{\boldsymbol{w}u}^{(t)}$ and $\beta_{\boldsymbol{\mu}u}^{(t)}(\nu_0, i)$ are parameters determining how much emphasis is put on the maximum value. The prior distribution is not informative (i.e., it is uniform) when $\beta_{\boldsymbol{w}u}^{(t)}$ and $\beta_{\boldsymbol{\mu}u}^{(t)}(\nu_0, i)$ are 0, corresponding to the case when no prior knowledge is available. In practice, however, $\beta_{\boldsymbol{\mu}u}^{(t)}(\nu_0, i)$ should not be 0 and a prior distribution of the tone model shapes should be provided. This is because if the prior distribution of the tone model shapes is not used, there are too many



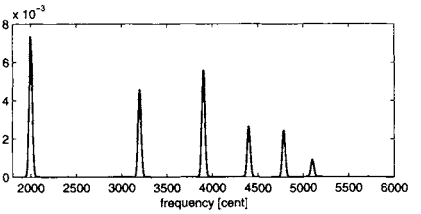
(a) for melody line
(u = 'melody', $i = 1, \nu_0 = 4000$ cent)



(b) for melody line
(u = 'melody', $i = 2, \nu_0 = 4000$ cent)



(c) for bass line
(u = 'bass line', $i = 1, \nu_0 = 2000$ cent)



(d) for bass line
(u = 'bass line', $i = 2, \nu_0 = 2000$ cent)

Fig. 11.4. Examples of prior distribution of the tone model shapes $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i))$.

degrees of freedom in their shapes. Without the prior distribution, unrealistic tone model shapes, such as a shape having only one salient component at frequency of the fourth harmonic component, could be estimated. In (11.15) and (11.16), $\mathsf{D}_{\boldsymbol{w}}(\boldsymbol{w}_{0u}^{(t)}; \boldsymbol{w}^{(t)})$ and $\mathsf{D}_{\boldsymbol{\mu}}(\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i); \boldsymbol{\mu}^{(t)}(\nu_0, i))$ are the following Kullback–Leibler information:

$$\mathsf{D}_{\boldsymbol{w}}(\boldsymbol{w}_{0u}^{(t)}; \boldsymbol{w}^{(t)}) = \int_{\mathrm{F}_u^{\mathrm{l}}}^{\mathrm{F}_u^{\mathrm{h}}} \sum_{i=1}^{\mathrm{I}_u} w_{0u}^{(t)}(\nu_0, i) \log \frac{w_{0u}^{(t)}(\nu_0, i)}{w^{(t)}(\nu_0, i)} d\nu_0, \tag{11.17}$$

$$\mathsf{D}_{\boldsymbol{\mu}}(\boldsymbol{\mu}_{0u}^{(t)}(\nu_0, i); \boldsymbol{\mu}^{(t)}(\nu_0, i)) = \sum_{m=1}^{M_u} c_{0u}^{(t)}(m|\nu_0, i) \log \frac{c_{0u}^{(t)}(m|\nu_0, i)}{c^{(t)}(m|\nu_0, i)}. \tag{11.18}$$

These prior distributions were originally introduced for the sake of analytical tractability of the expectation maximization (EM) algorithm to obtain intuitive (11.25) and (11.26).

MAP Estimation Using the EM Algorithm

The problem to be solved is to estimate the model parameter $\boldsymbol{\theta}^{(t)}$, taking into account the prior distribution $\mathsf{p}_{0u}(\boldsymbol{\theta}^{(t)})$, when $\mathsf{p}_{\Psi}^{(t)}(\nu)$ is observed. The MAP estimator of $\boldsymbol{\theta}^{(t)}$ is obtained by maximizing

$$\int_{-\infty}^{\infty} \mathsf{p}_{\Psi}^{(t)}(\nu)\,(\log \mathsf{p}(\nu|\boldsymbol{\theta}^{(t)}) + \log \mathsf{p}_{0u}(\boldsymbol{\theta}^{(t)}))\,d\nu. \tag{11.19}$$

Because this maximization problem is too difficult to solve analytically, the PreFEst core uses the expectation maximization (EM) algorithm (see the presentation of the EM algorithm in Chapter 2, p. 35 and [138]), which is an algorithm where two steps—the expectation step (E-step) and the maximization step (M-step)—are iteratively applied to compute MAP estimates from incomplete observed data (i.e., from $\mathsf{p}_{\Psi}^{(t)}(\nu)$). With respect to $\boldsymbol{\theta}^{(t)}$, each iteration updates the old estimate $\boldsymbol{\theta}'^{(t)} = \{w'^{(t)}, \boldsymbol{\mu}'^{(t)}\}$ to obtain a new (improved) estimate $\widehat{\boldsymbol{\theta}}^{(t)} = \{\widehat{w}^{(t)}, \widehat{\boldsymbol{\mu}}^{(t)}\}$. For each frame t, $w'^{(t)}$ is initialized with the final estimate $\widehat{w}^{(t-1)}$ after iterations at the previous frame $t-1$; $\boldsymbol{\mu}'^{(t)}$ is initialized with the most probable parameter $\boldsymbol{\mu}_{0u}^{(t)}$ in the current implementation.

By introducing the hidden (unobservable) variables ν_0, i, and m, which, respectively, describe which F0, which tone model, and which harmonic component were responsible for generating each observed frequency component at ν, the two steps can be specified as follows:

1. E-step:
   Compute the following $Q_{\mathrm{MAP}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)})$ for the MAP estimation:

$$Q_{\mathrm{MAP}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)}) = Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)}) + \log \mathsf{p}_{0u}(\boldsymbol{\theta}^{(t)}), \tag{11.20}$$

$$Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)}) = \int_{-\infty}^{\infty} \mathsf{p}_{\Psi}^{(t)}(\nu)\mathbb{E}_{\nu_0,i,m}[\log \mathsf{p}(\nu,\nu_0,i,m|\boldsymbol{\theta}^{(t)}) \mid \nu, \boldsymbol{\theta}'^{(t)}]\,d\nu, \tag{11.21}$$

   where $Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)})$ is the conditional expectation of the mean log-likelihood for the maximum likelihood estimation. $\mathbb{E}_{\nu_0,i,m}[a|b]$ denotes the conditional expectation of a with respect to the hidden variables ν_0, i, and m, with the probability distribution determined by condition b.
2. M-step:
   Maximize $Q_{\mathrm{MAP}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)})$ as a function of $\boldsymbol{\theta}^{(t)}$ to obtain an updated (improved) estimate $\widehat{\boldsymbol{\theta}}^{(t)}$:

$$\widehat{\boldsymbol{\theta}}^{(t)} = \operatorname*{argmax}_{\boldsymbol{\theta}^{(t)}} Q_{\mathrm{MAP}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)}). \tag{11.22}$$

   In the E-step, $Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)})$ is expressed as

$$Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'^{(t)}) = \int_{-\infty}^{\infty}\int_{\mathrm{F}_u^{\mathrm{l}}}^{\mathrm{F}_u^{\mathrm{h}}} \sum_{i=1}^{\mathrm{I}_u}\sum_{m=1}^{M_u} \mathsf{p}_{\Psi}^{(t)}(\nu) \times \mathsf{p}(\nu_0,i,m|\nu,\boldsymbol{\theta}'^{(t)}) \log \mathsf{p}(\nu,\nu_0,i,m|\boldsymbol{\theta}^{(t)})\,d\nu_0 d\nu, \tag{11.23}$$

where the complete-data log-likelihood is given by

$$\log \mathsf{p}(\nu, \nu_0, i, m|\boldsymbol{\theta}^{(t)}) = \log(w^{(t)}(\nu_0, i)\ \mathsf{p}(\nu, m|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))). \tag{11.24}$$

Regarding the M-step, (11.22) is a conditional problem of variation, where the conditions are given by (11.7) and (11.12). This problem can be solved by using Euler–Lagrange differential equations with Lagrange multipliers [222], [228] and the following new parameter estimates are obtained:

$$\widehat{w}^{(t)}(\nu_0, i) = \frac{\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i) + \beta^{(t)}_{\boldsymbol{wu}} w^{(t)}_{0u}(\nu_0, i)}{1 + \beta^{(t)}_{\boldsymbol{wu}}}, \tag{11.25}$$

$$\widehat{c}^{(t)}(m|\nu_0, i) = \frac{\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i)\ \widehat{c}^{(t)}_{\mathrm{ML}}(m|\nu_0, i) + \beta^{(t)}_{\boldsymbol{\mu u}}(\nu_0, i) c^{(t)}_{0u}(m|\nu_0, i)}{\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i) + \beta^{(t)}_{\boldsymbol{\mu u}}(\nu_0, i)}, \tag{11.26}$$

where $\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i)$ and $\widehat{c}^{(t)}_{\mathrm{ML}}(m|\nu_0, i)$ are, when the noninformative prior distribution ($\beta^{(t)}_{\boldsymbol{wu}} = 0$ and $\beta^{(t)}_{\boldsymbol{\mu u}}(\nu_0, i) = 0$) is given, the following maximum likelihood estimates:

$$\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i) = \int_{-\infty}^{\infty} \mathsf{p}^{(t)}_{\Psi}(\nu) \frac{w'^{(t)}(\nu_0, i)\ \mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}'^{(t)}(\nu_0, i))}{\int_{\mathrm{F}^{\mathrm{l}}_u}^{\mathrm{F}^{\mathrm{h}}_u} \sum_{k=1}^{\mathrm{I}_u} w'^{(t)}(\eta, k)\ \mathsf{p}(\nu|\eta, k, \boldsymbol{\mu}'^{(t)}(\eta, k))\, d\eta}\, d\nu, \tag{11.27}$$

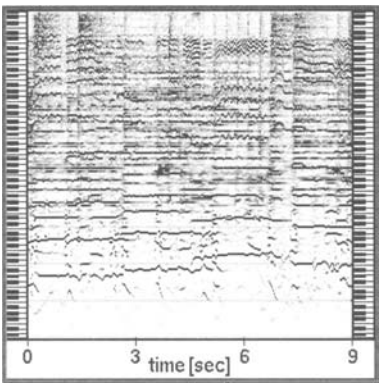
$$\widehat{c}^{(t)}_{\mathrm{ML}}(m|\nu_0, i) = \frac{1}{\widehat{w}^{(t)}_{\mathrm{ML}}(\nu_0, i)} \times \int_{-\infty}^{\infty} \mathsf{p}^{(t)}_{\Psi}(\nu) \frac{w'^{(t)}(\nu_0, i)\ \mathsf{p}(\nu, m|\nu_0, i, \boldsymbol{\mu}'^{(t)}(\nu_0, i))}{\int_{\mathrm{F}^{\mathrm{l}}_u}^{\mathrm{F}^{\mathrm{h}}_u} \sum_{k=1}^{\mathrm{I}_u} w'^{(t)}(\eta, k)\ \mathsf{p}(\nu|\eta, k, \boldsymbol{\mu}'^{(t)}(\eta, k))\, d\eta}\, d\nu. \tag{11.28}$$

After the above iterative computation of (11.25) and (11.26),[3] the F0s pdf $\mathsf{p}^{(t)}_{F0}(\nu_0)$ can be obtained from $w^{(t)}(\nu_0, i)$ according to (11.13). The tone model shape $c^{(t)}(m|\nu_0, i)$, which is the relative amplitude of each harmonic component of all types of tone models $\mathsf{p}(\nu|\nu_0, i, \boldsymbol{\mu}^{(t)}(\nu_0, i))$, can also be obtained.

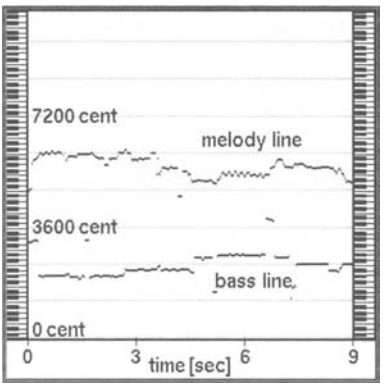
11.2.3 PreFEst Back End: Sequential F0 Tracking by Multiple-Agent Architecture

A simple way to identify the most predominant F0 is to find the frequency that maximizes the F0s pdf. This result is not always stable, however, because peaks corresponding to the F0s of simultaneous sounds sometimes compete in the F0s pdf for a moment and are transiently selected, one after another, as the maximum.

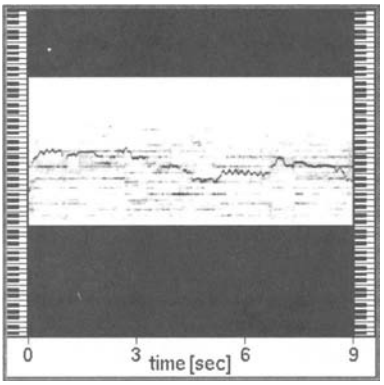
[3] In implementing the PreFEst core, this iterative computation is simple enough to perform only (11.25), (11.26), (11.27), and (11.28).



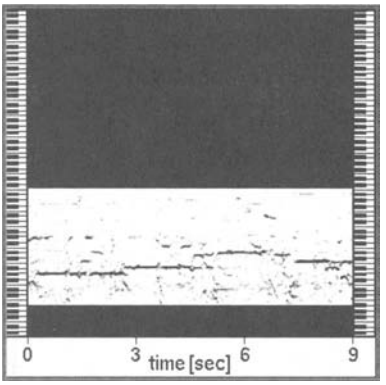
(a) Frequency components (observed pdf $p_{\Psi}^{(t)}(\nu)$ before applying bandpass filters)



(b) Estimated melody and bass lines (the most dominant and stable F0 trajectory in each $p_{F0}^{(t)}(\nu_0)$)



(c) F0s pdf ($p_{F0}^{(t)}(\nu_0)$) for estimating the melody line in (b)



(d) F0s pdf ($p_{F0}^{(t)}(\nu_0)$) for estimating the bass line in (b)

Fig. 11.5. Audio-synchronized real-time graphics output for a popular music excerpt with drum sounds: (a) frequency components, (b) the corresponding melody and bass lines estimated (final output), (c) the corresponding F0s pdf obtained when estimating the melody line, and (d) the corresponding F0s pdf obtained when estimating the bass line. These interlocking windows have the same vertical axis of log-scale frequency.

The PreFEst back end therefore considers the global temporal continuity of the F0 by using a multiple-agent architecture in which agents track different temporal trajectories of the F0 [228]. Each agent starts tracking from each salient peak in the F0s pdf, keeps tracking as long as it is temporally continued, and stops tracking when its next peak cannot be found for a while. The final F0 output is determined on the basis of the most dominant and stable F0 trajectory. Figure 11.5 shows an example of the final output.

11.2.4 Other Methods

While the PreFEst method resulted from pioneering research regarding melody and bass estimation and weighted-mixture modelling for F0 estimation, many issues still need to be resolved. For example, if an application requires MIDI-level note sequences of the melody line, the F0 trajectory should be segmented and organized into notes. Note that the PreFEst method does not deal with the problem of detecting the absence of melody and bass lines: it simply outputs the predominant F0 for every frame. In addition, since the melody and bass lines are generated from a process that is statistically biased rather than random—i.e., their transitions are musically appropriate this bias can also be incorporated into their estimation. This section introduces other recent approaches [494], [493], [435], [436], [169] that deal with these issues in describing polyphonic audio signals.

Paiva, Mendes, and Cardoso [494], [493] proposed a method of obtaining the melody note sequence by using a model of the human auditory system [595] as a frequency-analysis front end and applying MIDI-level note tracking, segmentation, and elimination techniques. Although the techniques used differ from the PreFEst method, the basic idea that 'the melody generally clearly stands out of the background' is the same as the basic PreFEst concept that the F0 of the most predominant harmonic structure is considered the melody. The advantage of this method is that MIDI-level note sequences of the melody line are generated, while the output of PreFEst is a simple temporal trajectory of the F0. The method first estimates predominant F0 candidates by using correlograms (see Chapter 8) that represent the periodicities in a cochleagram (auditory nerve responses of an ear model). It then forms the temporal trajectories of F0 candidates: it quantizes their frequencies to the closest MIDI note numbers and then tracks them according to their frequency proximity, where only one-semitone transition is considered continuous. After this tracking, F0 trajectories are segmented into MIDI-level note candidates by finding a sufficiently long trajectory having the same note number and by dividing it at clear local minima of its amplitude envelope. Because there still remain many inappropriate notes, it eliminates notes whose amplitude is too low, whose duration is too short, or which have harmonically related F0s and almost same onset and offset times. Finally, the melody note sequence is obtained by selecting the most predominant notes according to heuristic rules. Since simultaneous notes are not allowed, the method eliminates simultaneous notes that are less dominant and not in a middle frequency range.

Marolt [435], [436] proposed a method of estimating the melody line by representing it as a set of short vocal fragments of F0 trajectories. This method is based on the PreFEst method with some modifications: it uses the PreFEst core to estimate predominant F0 candidates, but uses a spectral modelling synthesis (SMS) front end that performs the sinusoidal modelling and analysis (see Chapters 1 and 3) instead of the PreFEst front end. The advantage of this method is that the F0 candidates are tracked and grouped into melodic

fragments (reasonably segmented signal regions that exhibit strong and stable F0) and these fragments are then clustered into the melody line. The method first tracks temporal trajectories of the F0 candidates (salient peaks) to form the melodic fragments by using a salient peak tracking approach similar to the PreFEst back end (though it does not use multiple agents). Because the fragments belong to not only the melody (lead vocal), but also to different parts of the accompaniment, they are clustered to find the melody cluster by using Gaussian mixture models (GMMs) according to their five properties:

- Dominance (average weight of a tone model estimated by the EM algorithm),
- Pitch (centroid of the F0s within the fragment),
- Loudness (average loudness of harmonics belonging to the fragment),
- Pitch stability (average change of F0s during the fragment), and
- Onset steepness (steepness of overall loudness change during the first 50 ms of the fragment).

Eggink and Brown [169] proposed a method of estimating the melody line with the emphasis on using various knowledge sources, such as knowledge about instrument pitch ranges and interval transitions, to choose the most likely succession of F0s as the melody line. Unlike other methods, this method is specialized for a classical sonata or concerto, where a solo melody instrument can span the whole pitch range, ranging from the low tones of a cello to a high-pitched flute, so the frequency range limitation used in the PreFEst method is not feasible. In addition, because the solo instrument does not always have the most predominant F0, additional knowledge sources are necessary to extract the melody line. The main advantage of this method is the leverage provided by knowledge sources, including local knowledge about an instrument recognition module and temporal knowledge about tone durations and interval transitions, which are integrated in a probabilistic search. Those sources can both help to choose the correct F0 among multiple concurrent F0 candidates and to determine sections where the solo instrument is actually present. The knowledge sources consist of two categories, local knowledge and temporal knowledge. The local knowledge concerning F0 candidates obtained by picking peaks in the spectrum includes

- F0 strength (the stronger the spectral peak, the higher its likelihood of being the melody),
- Instrument-dependent F0 likelihood (the likelihood values of an F0 candidate in terms of its frequency and the pitch range of each solo instrument, which are evaluated by counting the frequency of its F0 occurrence in different standard MIDI files), and
- Instrument likelihood (the likelihood values of an F0 candidate being produced by each solo instrument, which are evaluated by the instrument recognition module).

The instrument recognition module uses trained Gaussian classifiers of the frequency and power of the first ten harmonic components, their deltas, and their delta-deltas, which are taken from the spectrum for each F0 candidate. On the other hand, the temporal knowledge concerning tone candidates obtained by connecting F0 candidates includes

- Instrument-dependent interval likelihood (the likelihood values of an interval transition between succession tones, which are evaluated by counting the frequency of its interval occurrence in different standard MIDI files), and
- Relative tone usage (measures related to tone durations between successive tones, which are used to penalize overlapped tones).

These knowledge sources are combined to find the most likely 'path' of the melody through the space of all F0 candidates in time. Since the melody path occasionally follows the accompaniment, additional postprocessing is done to eliminate sections where the solo instrument is actually silent.

While the above methods deal with the melody line, Hainsworth and Macleod [264] proposed a method of obtaining the bass note sequence by maintaining multiple hypotheses. The method first extracts the onset times of bass notes by picking peaks of a smoothed temporal envelope of a total power below 200 Hz. It then generates hypotheses regarding the F0 of each extracted note; the F0 and amplitude of each hypothesis are estimated by fitting a quadratic polynomial to a large amplitude peak and subtracting it from the spectrum. The first four harmonic components of those hypotheses are tracked over time by using a comb-filter-like analysis. Finally, the method selects the most likely hypothesis for each onset on the basis of its duration and the amplitude of harmonic components and further tidies up these hypotheses by removing inappropriate overlaps and relatively low amplitude notes.

11.3 Estimating Beat Structure

Beat tracking (including measure or bar line estimation) is defined as the process of organizing musical audio signals into a hierarchical beat structure (including beat and measure levels). It is also an important initial step in the computational modelling of music understanding because the beat is fundamental, for both trained and untrained listeners, to the perception of Western music. As described in Section 11.7.2 and Section 4.1, p. 101, there are many applications such as music-synchronized computer graphics, stage lighting control, video/audio synchronization, and human–computer improvisation in live ensembles.

Various methods for estimating the beat structure are described in detail in Chapter 4. Here, the synergy between the estimation of the hierarchical beat structure, drum patterns, and chord changes is briefly discussed. This synergy is exploited in a real-time beat-tracking system developed by Goto and

Muraoka [235], [220], [221]. The estimation of the hierarchical beat structure, especially the measure (bar line) level, requires the use of musical knowledge about drum patterns and chord changes; on the other hand, drum patterns and chord changes are difficult to estimate without referring to the beat structure of the beat level (quarter note level). The system addresses this issue by leveraging the integration of top-down and bottom-up processes (Fig. 11.6) under the assumption that the time signature of an input song is 4/4. The system first obtains multiple possible hypotheses of provisional beat times (quarter-note-level beat structure) on the basis of onset times without using musical knowledge about drum patterns and chord changes. Because the onset times of the sounds of bass drum and snare drum can be detected by a bottom-up frequency analysis described in Section 5.2.3, p. 137, the system makes use of the provisional beat times as top-down information to form the detected onset times into drum patterns whose grid is aligned with the beat times. The system also makes use of the provisional beat times to detect chord changes in a frequency spectrum without identifying musical notes or chords by name. The frequency spectrum is sliced into strips at the beat times and the dominant frequencies of each strip are estimated by using a histogram of frequency components in the strip [240]. Chords are considered to be changed when the dominant frequencies change between adjacent strips. After the drum patterns and chord changes are obtained, the higher-level beat structure, such as the measure level, can be estimated by using musical knowledge regarding them.

11.4 Estimating Drums

The detection of the onset times of drum sounds is important because the basic rhythms of popular music pieces including drum sounds are mainly characterized by drum performances. As described in Section 11.7.1, there are many applications such as rhythm-based music information retrieval and genre classification.

Various methods for detecting drum sounds are described in detail in Chapter 5.

11.5 Estimating Chorus Sections and Repeated Sections

Chorus ('hook' or 'refrain') sections of popular music are the most representative, uplifting, and prominent thematic sections in the music structure of a song, and human listeners can easily understand where the chorus sections are because these sections are the most repeated and memorable portions of a song. Automatic detection of chorus sections is essential for the computational modelling of music understanding and is useful in various practical applications. In music browsers or music retrieval systems, it enables a listener to quickly preview a chorus section as a 'music thumbnail' (a musical equivalent

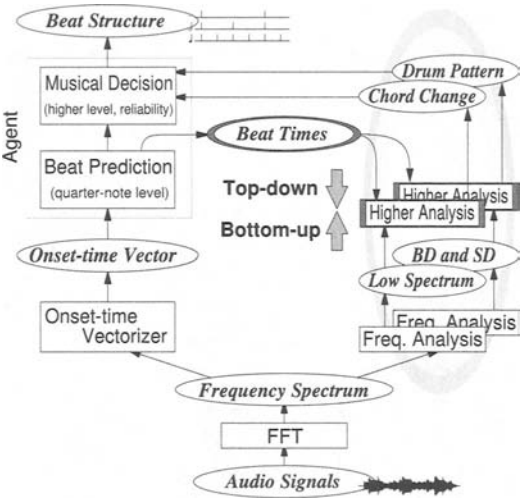


Fig. 11.6. Synergy between the estimation of the hierarchical beat structure, drum patterns, and chord changes. Drum patterns and chord changes are obtained, at 'Higher Analysis' in the figure, by using provisional beat times as top-down information. The hierarchical beat structure is then estimated, at 'Musical Decision' in the figure, by using the drum patterns and chord changes. A drum pattern is represented by the temporal pattern of a bass drum (BD) and a snare drum (SD).

of an image thumbnail) to find a desired song. It can also provide novel music listening interfaces for end users as described in Section 11.7.3.

To detect chorus sections, typical approaches do not rely on prior information regarding acoustic features unique to choruses but focus on the fact that chorus sections are usually the most repeated sections of a song. They thus adopt the following basic strategy: detect similar sections that repeat within a musical piece (such as a repeating phrase) and output those that appear most often. On entering the 2000s, this strategy has led to methods for extracting a single segment from several chorus sections by detecting a repeated section of a designated length as the most representative part of a musical piece [417], [27], [103]; methods for segmenting music, discovering repeated structures, or summarizing a musical piece through bottom-up analyses without assuming the output segment length [110], [111], [512], [516], [23], [195], [104], [82], [664], [420]; and a method for exhaustively detecting all chorus sections by determining the start and end points of every chorus section [224].

Although this basic strategy of finding sections that repeat most often is simple and effective, it is difficult for a computer to judge repetition because it is rare for repeated sections to be exactly the same. The following summarizes the main problems that must be addressed in finding music repetition and determining chorus sections.

Problem 1: Extracting acoustic features and calculating their similarity
Whether a section is a repetition of another must be judged on the basis of the similarity between the acoustic features obtained from each frame or section. In this process, the similarity must be high between acoustic features even if the accompaniment or melody line changes somewhat in the repeated section (e.g., the absence of accompaniment on bass and/or drums after repetition). That is, it is necessary to use features that capture useful and invariant properties.

Problem 2: Finding repeated sections
A pair of repeated sections can be found by detecting contiguous temporal regions having high similarity. However, the criterion establishing how high similarity must be to indicate repetition depends on the song. For a song in which repeated accompaniment phrases appear very often, for example, only a section with very high similarity should be considered the chorus section repetition. For a song containing a chorus section with accompaniments changed after repetition, on the other hand, a section with somewhat lower similarity can be considered the chorus section repetition. This criterion can be easily set for a small number of specific songs by manual means. For a large open song set, however, the criterion should be automatically modified based on the song being processed.

Problem 3: Grouping repeated sections
Even if many pairs of repeated sections with various lengths are obtained, it is not obvious how many times and where a section is repeated. It is therefore necessary to organize repeated sections that have common sections into a group. Both ends (the start and end points) of repeated sections must also be estimated by examining the mutual relationships among various repeated sections. For example, given a song having the structure (A B C B C C), the long repetition corresponding to (B C) would be obtained by a simple repetition search. Both ends of the C section in (B C) could be inferred, however, from the information obtained regarding the final repetition of C in this structure.

Problem 4: Detecting modulated repetition
Because the acoustic features of a section generally undergo a significant change after modulation (key change; see Section 1.1, p. 7), similarity with the section before modulation is low, making it difficult to judge repetition. The detection of modulated repetition is important since modulation sometimes occurs in chorus repetitions, especially in the latter half of a song.[4]

Problem 5: Selecting chorus sections
Because various levels of repetition can be found in a musical piece, it is necessary to select a group of repeated sections corresponding to chorus

[4]Masataka Goto's survey of Japan's popular music hit chart (top 20 singles ranked weekly from 2000 to 2003) showed that modulation occurred in chorus repetitions in 152 songs (10.3%) out of 1481.

sections. A simple selection of the most repeated sections is not always appropriate though. For example, another section such as verse A is occasionally repeated more often than chorus sections.

Regarding the above repetition-based methods, the following sections mainly describe a method called *RefraiD* (Refrain Detection method) [224] and briefly introduce techniques used in the other methods in each relevant section and Section 11.5.6. Since the RefraiD method addresses all of the above problems and detects all chorus sections in a popular music song regardless of whether a key change occurs, it is suitable for music scene description. Figure 11.7 shows the process flow of the RefraiD method. First, a 12-dimensional feature vector called a *chroma vector*, which is robust with respect to changes of accompaniments, is extracted from each frame of an input audio signal and then the similarity between these vectors is calculated *(solution to Problem 1)*. Each element of the chroma vector corresponds to one of the 12 pitch classes (C, C#, D, ..., B) and is the sum of the magnitude spectrum at frequencies of its pitch class over six octaves. Pairs of repeated sections are then listed (found) using an adaptive repetition-judgement criterion which is configured by an automatic threshold selection method based on a discriminant criterion *(solution to Problem 2)*. To organize common repeated sections into groups and to identify both ends of each section, the pairs of repeated sections are integrated (grouped) by analysing their relationships over the whole song *(solution to Problem 3)*. Because each element of a chroma vector corresponds to a different pitch class, a before-modulation chroma vector is close to the after-modulation chorus vector whose elements are shifted (exchanged) by the pitch difference of the key change. By considering 12 kinds of shift (pitch differences), 12 sets of the similarity between non-shifted and shifted chroma vectors are then calculated, pairs of repeated sections from those sets are listed, and all of them are integrated *(solution to Problem 4)*. Finally, the *chorus measure*, which is the possibility of being chorus sections for each group, is evaluated *(solution to Problem 5)*, and the group of chorus sections with the highest chorus measure as well as other groups of repeated sections are output (Fig. 11.8).

11.5.1 Extracting Acoustic Features and Calculating Their Similarity

The following acoustic features, which capture pitch and timbral features of audio signals in different ways, were used in various methods: chroma vectors [224], [27], [110], [111], mel-frequency cepstral coefficients (MFCC) [417], [103], [23], [195], [104], (dimension-reduced) spectral coefficients [103], [195], [104], [82], [664], pitch representations using F0 estimation or constant-Q filterbanks [110], [111], [82], [420], and dynamic features obtained by supervised learning [512], [516].

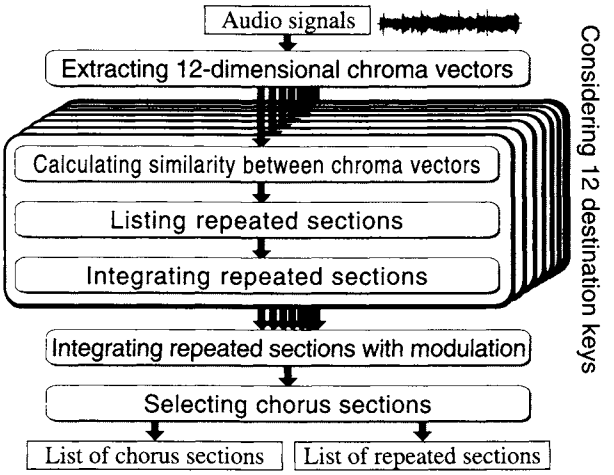


Fig. 11.7. Overview of RefraiD (Refrain Detection method) for detecting all chorus sections with their start and end points while considering modulations (key changes).

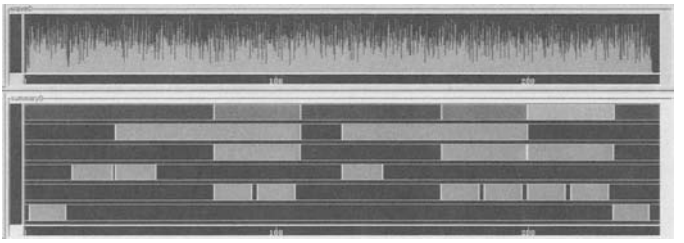


Fig. 11.8. An example of chorus sections and repeated sections detected by the RefraiD method. The horizontal axis is the time axis (in seconds) covering the entire song. The upper window shows the power. On each row in the lower window, coloured sections indicate similar (repeated) sections. The top row shows the list of the detected chorus sections, which were correct for this song (RWC-MDB-P-2001 No. 18 of the RWC Music Database [229], [227]) and the last of which was modulated. The bottom five rows show the list of various repeated sections (only the five longest repeated sections are shown). For example, the second row from the top indicates the structural repetition of 'verse A ⇒ verse B ⇒ chorus'; the bottom row with two short coloured sections indicates the similarity between the 'intro' and 'ending'.

Pitch Feature: Chroma Vector

The chroma vector is a perceptually motivated feature vector using the concept of *chroma* in Shepard's helix representation of musical pitch perception [584]. According to Shepard [584], the perception of pitch with respect to a musical context can be graphically represented by using a continually cyclic helix that has two dimensions, *chroma* and *height*, as shown at the right of Fig. 11.9. Chroma refers to the position of a musical pitch within an octave

that corresponds to a cycle of the helix; i.e., it refers to the position on the circumference of the helix seen from directly above. On the other hand, height refers to the vertical position of the helix seen from the side (the position of an octave).

Figure 11.9 shows an overview of calculating the chroma vector used in the RefraiD method [224]. This represents magnitude distribution on the chroma that is discretized into twelve pitch classes within an octave. The 12-dimensional chroma vector $\mathbf{v}(t)$ is extracted from the magnitude spectrum, $\Psi_p(\nu, t)$ at the log-scale frequency ν at time t, calculated by using the short-time Fourier transform (STFT). Each element of $\mathbf{v}(t)$ corresponds to a pitch class c ($c = 1, 2, \ldots, 12$) in the equal temperament and is represented as $v_c(t)$:

$$v_c(t) = \sum_{h=\mathrm{Oct_L}}^{\mathrm{Oct_H}} \int_{-\infty}^{\infty} \mathsf{BPF}_{c,h}(\nu)\, \Psi_p(\nu, t)\, d\nu. \tag{11.29}$$

The $\mathsf{BPF}_{c,h}(\nu)$ is a bandpass filter that passes the signal at the log-scale centre frequency $F_{c,h}$ (in cents) of pitch class c (chroma) in octave position h (height), where

$$F_{c,h} = 1200h + 100(c - 1). \tag{11.30}$$

The $\mathsf{BPF}_{c,h}(\nu)$ is defined using a Hanning window as follows:

$$\mathsf{BPF}_{c,h}(\nu) = \frac{1}{2}\left(1 - \cos\frac{2\pi(\nu - (F_{c,h} - 100))}{200}\right), \quad \nu \in [0, 200]. \tag{11.31}$$

This filter is applied to octaves from $\mathrm{Oct_L}$ to $\mathrm{Oct_H}$. In Goto's implementation [224], an STFT with a 256 ms Hanning window[5] shifted by 80 ms is calculated for audio signals sampled at 16 kHz, and the $\mathrm{Oct_L}$ and $\mathrm{Oct_H}$ are respectively 3 and 8, covering six octaves (130 Hz to 8 kHz).

There are variations in how the chroma vector is calculated. For example, Bartsch and Wakefield [27] developed a technique where each STFT bin of the log-magnitude spectrum is mapped directly to the most appropriate pitch class, and Dannenberg and Hu [110], [111] also used this technique. A similar continuous concept was called the chroma spectrum [655].

There are several advantages to using the chroma vector. Because it captures the overall harmony (pitch-class distribution), it can be similar even if accompaniments or melody lines are changed to some degree after repetition. In fact, the chroma vector is effective for identifying chord names [201], [678], [679], [583], [684]. The chroma vector also enables modulated repetition to be detected as described in Section 11.5.4.

Timbral Feature: MFCC and Dynamic Features

While the chroma vectors capture pitch-related content, the MFCCs (see Section 2.1.3, p. 25 for a presentation of MFCCs) typically used in speech

[5]The window length is determined to obtain good frequency resolution in a low-frequency region.

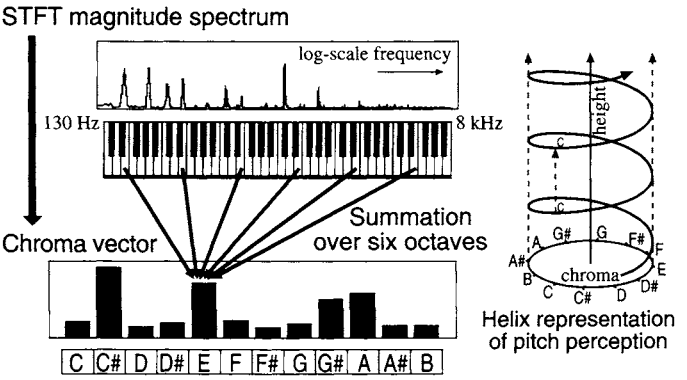


Fig. 11.9. Overview of calculating a 12-dimensional chroma vector. The magnitude at six different octaves is summed into just one octave which is divided into 12 log-spaced divisions corresponding to pitch classes. Shepard's helix representation of musical pitch perception [584] is shown at the right.

recognition capture spectral content and general pitch range, and are useful for finding timbral or 'texture' repetitions. Dynamic features [512], [516] are more adaptive spectral features that are designed for music structure discovery through a supervised learning method. Those features are selected from the spectral coefficients of a filterbank output by maximizing the mutual information between the selected features and hand-labelled music structures. The dynamic features are beneficial in that they reduce the size of the results when calculating similarity (i.e., the size of the similarity matrix described in Section 11.5.1) because the frame shift can be longer (e.g., 1 s) than for other features.

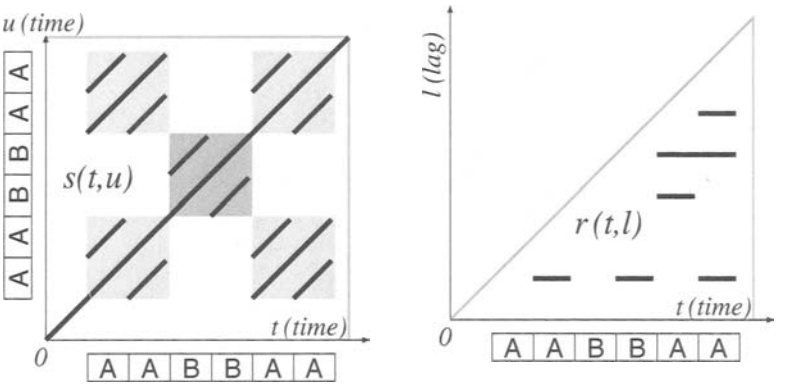
Calculating Similarity

Given a feature vector such as the chroma vector or MFCC at every frame, the next step is to calculate the similarity between feature vectors. Various distance or similarity measures, such as the Euclidean distance and the cosine angle (inner product), can be used for this. Before calculating the similarity, feature vectors are usually normalized, for example, to a mean of zero and a standard deviation of one or to a maximum element of one.

In the RefraiD method [224], the similarity $\mathsf{r}(t,l)$ between the feature vectors (chroma vectors) $\mathbf{v}(t)$ and $\mathbf{v}(t-l)$ is defined as

$$\mathsf{r}(t,l) = 1 - \frac{1}{\sqrt{12}} \left| \frac{\mathbf{v}(t)}{\max_c v_c(t)} - \frac{\mathbf{v}(t-l)}{\max_c v_c(t-l)} \right|, \tag{11.32}$$

where l is the lag and $v_c(t)$ is an element of $\mathbf{v}(t)$ (11.29). Since the denominator $\sqrt{12}$ is the length of the diagonal line of a 12-dimensional hypercube with edge length 1, $\mathsf{r}(t,l)$ satisfies $0 \leq \mathsf{r}(t,l) \leq 1$.



(a) Similarity matrix (b) Time-lag triangle

Fig. 11.10. An idealized example of a similarity matrix and time-lag triangle drawn from the same feature vectors of a musical piece consisting of four 'A' sections and two 'B' sections. The diagonal line segments in the similarity matrix or horizontal line segments in the time-lag triangle, which represent similar sections, appear when short-time pitch features like chroma vectors are used.

For the given 36-dimensional feature vectors of a constant-Q filterbank output with centre frequencies at 36 tempered semitones in 3 octaves, Lu, Wang, and Zhang [420] introduced an original distance measure that emphasizes melody similarity and suppresses timbre similarity. This measure does not depend on the norm of the difference between the 36-dimensional feature vectors, but on the structure of it. It considers how the peak intervals in the difference conform to harmonic relationships such as perfect fifth and octave.

11.5.2 Finding Repeated Sections

By using the same similarity measure $\mathsf{r}(t, l)$, two equivalent representations can be obtained: a *similarity matrix* [103], [110], [111], [195], [104], [664] and a *time-lag triangle* (or time-lag matrix) [224], [27], [516], [420], as shown in Fig. 11.10. For the similarity matrix, the similarity $\mathsf{s}(t, u)$ between feature vectors $\mathbf{v}(t)$ and $\mathbf{v}(u)$,

$$\mathsf{s}(t, u) = \mathsf{r}(t, t - u), \tag{11.33}$$

is drawn within a square in the two-dimensional (t-u) space.[6] For the time-lag triangle, the similarity $\mathsf{r}(t, l)$ between feature vectors $\mathbf{v}(t)$ and $\mathbf{v}(t - l)$ is drawn within a right-angled isosceles triangle in the two-dimensional time-lag (t-l) space. If a nearly constant tempo can be assumed, each pair of similar sections is represented by two non-central diagonal line segments in the

[6] As described in Section 4.6, p. 112, the similarity matrix can also be used to examine rhythmic structure.

similarity matrix or a horizontal line segment in the time-lag triangle. Because the actual $\mathsf{r}(t,l)$ obtained from a musical piece is noisy and ambiguous, it is not a straightforward task to detect these line segments.

The RefraiD method [224] finds all horizontal line segments (contiguous regions with high $\mathsf{r}(t,l)$) in the time-lag triangle by evaluating $\mathsf{R}_{\mathrm{all}}(t,l)$, the possibility of containing line segments at the lag l at the current time t (e.g., at the end of a song[7]) as follows (Fig. 11.11):[8]

$$\mathsf{R}_{\mathrm{all}}(t,l) = \frac{1}{t-l+1}\sum_{\tau=l}^{t}\mathsf{r}(\tau,l). \tag{11.34}$$

Before this calculation, $\mathsf{r}(t,l)$ is normalized by subtracting a local mean value while removing noise and emphasizing horizontal lines. In more detail, given each point $\mathsf{r}(t,l)$ in the time-lag triangle, six-directional local mean values along the right, left, upper, lower, upper right, and lower left directions starting from the point $\mathsf{r}(t,l)$ are calculated, and the maximum and minimum are obtained. If the local mean along the right or left direction takes the maximum, $\mathsf{r}(t,l)$ is considered part of a horizontal line and emphasized by subtracting the minimum from $\mathsf{r}(t,l)$. Otherwise, $\mathsf{r}(t,l)$ is considered noise and suppressed by subtracting the maximum from $\mathsf{r}(t,l)$; noise tends to appear as lines along the upper, lower, upper right, and lower left directions.

The method then picks up each peak in $\mathsf{R}_{\mathrm{all}}(t,l)$ along the lag l after smoothing $\mathsf{R}_{\mathrm{all}}(t,l)$ with a moving average filter along the lag and removing a global drift (bias) caused by cumulative noise in $\mathsf{r}(t,l)$[9] from $\mathsf{R}_{\mathrm{all}}(t,l)$. The method next selects only high peaks above a threshold to search the line segments. Because this threshold is closely related to the repetition-judgement criterion which should be adjusted for each song, an automatic threshold selection method based on a discriminant criterion [491] is used. When dichotomizing the peak heights into two classes by a threshold, the optimal threshold is obtained by maximizing the discriminant criterion measure defined by the following between-class variance:

$$\sigma_B^2 = \omega_1\omega_2(\mu_1-\mu_2)^2, \tag{11.35}$$

where ω_1 and ω_2 are the probabilities of class occurrence (number of peaks in each class/total number of peaks), and μ_1 and μ_2 are the means of the peak heights in each class.

[7] $\mathsf{R}_{\mathrm{all}}(t,l)$ is evaluated along with the real-time audio input for a real-time system based on RefraiD. On the other hand, it is evaluated at the end of a song for a non-real-time off-line analysis.

[8] This can be considered the Hough transform where only horizontal lines are detected: the parameter (voting) space $\mathsf{R}_{\mathrm{all}}(t,l)$ is therefore simply one dimensional along l.

[9] Because the similarity $\mathsf{r}(\tau,l)$ is noisy, its sum $\mathsf{R}_{\mathrm{all}}(t,l)$ tends to be biased: the longer the summation period for $\mathsf{R}_{\mathrm{all}}(t,l)$, the higher the summation result by (11.34).

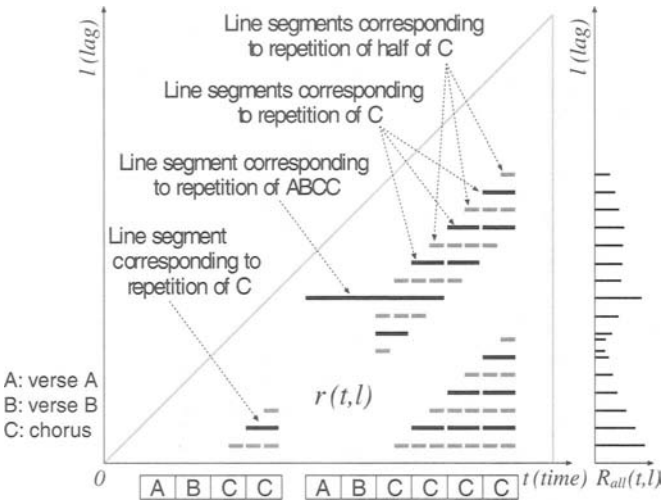


Fig. 11.11. A sketch of line segments, the similarity $\mathsf{r}(t,l)$ in the time-lag triangle, and the possibility $\mathsf{R}_{\mathrm{all}}(t,l)$ of containing line segments at lag l.

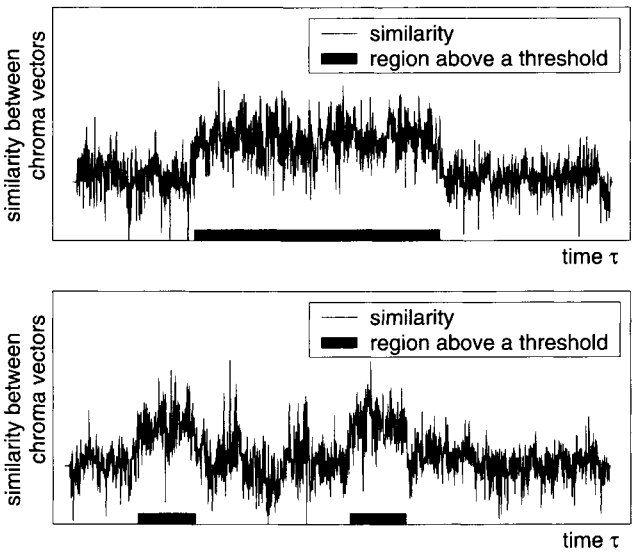


Fig. 11.12. Examples of the similarity $\mathsf{r}(\tau, l_1)$ at high-peak lags l_1. The bottom horizontal bars indicate the regions above an automatically adjusted threshold, which means they correspond to line segments.

For each picked-up high peak with lag l_1, the line segments are finally searched on the one-dimensional function $\mathsf{r}(\tau, l_1)$ $(l_1 \leq \tau \leq t)$. After smoothing $\mathsf{r}(\tau, l_1)$ using a moving average filter, the method obtains line segments on

which the smoothed $\mathsf{r}(\tau, l_1)$ is above a threshold (Fig. 11.12). This threshold is also adjusted through the automatic threshold selection method.

Instead of using the similarity matrix and time-lag triangle, there are other approaches that do not explicitly find repeated sections. To segment music, represent music as a succession of states (labels), and obtain a music thumbnail or summary, these approaches segment and label (i.e., categorize) contiguous frames (feature vectors) by using clustering techniques [417] or ergodic hidden Markov models (HMMs) [417], [512], [516] (HMMs are introduced on p. 63 of this volume).

11.5.3 Grouping Repeated Sections

Since each line segment in the time-lag triangle indicates just a pair of repeated sections, it is necessary to organize into a group the line segments that have common sections—i.e., overlap in time. When a section is repeated N times ($N \geq 3$), the number of line segments to be grouped together should theoretically be $N(N-1)/2$ if all of them are found in the time-lag triangle.

Aiming to exhaustively detect all the repeated (chorus) sections appearing in a song, the RefraiD method groups line segments having almost the same section while redetecting some missing (hidden) line segments not found in the bottom-up detection process (described in Section 11.5.2) through top-down processing using information on other detected line segments. In Fig. 11.11, for example, two line segments corresponding to the repetition of the first and third C and the repetition of the second and fourth C, which overlap with the long line segment corresponding to the repetition of ABCC, can be found even if they were hard to find in the bottom-up process. The method also appropriately adjusts the start and end times of line segments in each group because they are sometimes inconsistent in the bottom-up line segment detection.

11.5.4 Detecting Modulated Repetition

The processes described above do not deal with modulation (key change), but they can easily be extended to it. A modulation can be represented by the pitch difference of its key change, ζ $(0, 1, \ldots, 11)$, which denotes the number of tempered semitones. For example, $\zeta = 9$ means the modulation of nine semitones upward or the modulation of three semitones downward. One of the advantages of the 12-dimensional chroma vector $\mathbf{v}(t)$ is that a transposition amount ζ of the modulation can naturally correspond to the amount by which its 12 elements are shifted (rotated). When $\mathbf{v}(t)$ is the chroma vector of a certain performance and $\mathbf{v}(t)'$ is the chroma vector of the performance that is modulated by ζ semitones upward from the original performance, they tend to satisfy

$$\mathbf{v}(t) \approx \mathbf{S}^{\zeta}\mathbf{v}(t)', \tag{11.36}$$

where $\mathbf{S}$ is a 12-by-12 shift matrix defined by

$$\mathbf{S} = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}. \tag{11.37}$$

To detect modulated repetition by using this feature of chroma vectors and considering 12 destination keys, the RefraiD method [224] calculates 12 kinds of extended similarity as follows:

$$r_\varsigma(t,l) = 1 - \frac{1}{\sqrt{12}} \left| \frac{\mathbf{S}^\varsigma \mathbf{v}(t)}{\max_c v_c(t)} - \frac{\mathbf{v}(t-l)}{\max_c v_c(t-l)} \right|. \tag{11.38}$$

Starting from each $r_\varsigma(t,l)$, the processes of finding and grouping the repeated sections are performed again. Non-modulated and modulated repeated sections are then grouped if they share the same section.

11.5.5 Selecting Chorus Sections

A group corresponding to the chorus sections is finally selected from groups of repeated sections (line segments). In general, a group that has many and long repeated sections tends to be the chorus sections. In addition to this property, the RefraiD method evaluates the *chorus measure*, which is the possibility of being chorus sections for each group, by considering the following three heuristic rules with a focus on popular music:

1. The length of the chorus has an appropriate, allowed range (7.7 to 40 s in Goto's implementation).
2. When there is a repeated section that is long enough to likely correspond to the repetition of a long section like (verse A $\Rightarrow$ verse B $\Rightarrow$ chorus) $\times$ 2, the chorus section is likely to be at the end of that repeated section.
3. Because a chorus section tends to have two half-length repeated subsections within its section, a section having those subsections is likely to be the chorus section.

The group that maximizes the chorus measure is finally selected as the chorus sections.

11.5.6 Other Methods

Since the above sections mainly describe the RefraiD method [224] with the focus on detecting all chorus sections, this section briefly introduces other methods [417], [27], [103], [110], [111], [512], [516], [23], [195], [104], [82], [664],

[420] that aim at music thumbnailing, music segmentation, structure discovery, or music summarization.

Several methods for detecting the most representative part of a song for use as a music thumbnail have been studied. Logan and Chu [417] developed a method using clustering techniques and hidden Markov models (HMMs) to categorize short segments (1 s) in terms of their acoustic features, where the most frequent category is then regarded as a chorus. Bartsch and Wakefield [27] developed a method that calculates the similarity between acoustic features of beat-length segments obtained by beat tracking and finds the given-length segment with the highest similarity averaged over its segment. Cooper and Foote [103] developed a method that calculates a similarity matrix of acoustic features of short frames (100 ms) and finds the given-length segment with the highest similarity between it and the whole song. Note that these methods assume that the output segment length is given and do not identify both ends of a repeated section.

Music segmentation or structure discovery methods where the output segment length is not assumed have also been studied. Dannenberg and Hu [110], [111] developed a structure discovery method of clustering pairs of similar segments obtained by several techniques such as efficient dynamic programming or iterative greedy algorithms. This method finds, groups, and removes similar pairs from the beginning to group all the pairs. Peeters, La Burthe, and Rodet [512], [516] developed a supervised learning method of modelling dynamic features and studied two structure discovery approaches: the sequence approach of obtaining repetitions of patterns and the state approach of obtaining a succession of states. The dynamic features are selected from the spectrum of a filterbank output by maximizing the mutual information between the selected features and hand-labelled music structures. Aucouturier and Sandler [23] developed two methods for finding repeated patterns in a succession of states (texture labels) obtained by HMMs. They used two image processing techniques, the kernel convolution and Hough transform, to detect line segments in the similarity matrix between the states. Foote and Cooper [195], [104] developed a method of segmenting music by correlating a kernel along the diagonal of the similarity matrix, and clustering the obtained segments on the basis of the self-similarity of their statistics. Chai and Vercoe [82] developed a method of detecting segment repetitions by using dynamic programming, clustering the obtained segments, and labelling the segments based on heuristic rules such as the rule of first labelling the most frequent segments, removing them, and repeating the labelling process. Wellhausen and Crysandt [664] studied the similarity matrix of spectral envelope features defined in the MPEG-7 descriptors and a technique of detecting non-central diagonal line segments. Lu, Wang, and Zhang [420] developed a method of analysing all repeated sections by using a structure-based distance measure that emphasizes pitch similarity over timbral similarity. Their method also estimates the tempo of a song and discriminates between vocal and instrumental sections to facilitate music structure analysis.

11.6 Evaluation Issues

To evaluate automatic music scene description methods, it is necessary to label musical pieces in an adequate-size music database with their correct descriptions (metadata). This labelling task is time consuming and troublesome. More seriously, there was no available common music database with correct metadata since most musical pieces used by researchers are generally copyrighted and cannot be shared by other researchers.

But since 2000, a copyright-cleared music database, called the RWC (Real World Computing) Music Database [229], [230], [227], was developed and has been available to researchers as a common foundation for research. It contains six original collections: the Popular Music Database (100 pieces), Royalty-Free Music Database (15 pieces), Classical Music Database (50 pieces), Jazz Music Database (50 pieces), Music Genre Database (100 pieces), and Musical Instrument Sound Database (50 instruments). For all 315 musical pieces, audio signals, standard MIDI files, and text files of lyrics were prepared. For the 50 instruments, individual sounds at half-tone intervals were captured. This database has been distributed to researchers around the world and has already been widely used. For musical instrument sounds, there are other databases released for public use: the McGill University Master Samples [487] and the University of Iowa Musical Instrument Samples [198]. Musical pieces licensed under a Creative Commons license can also be used for evaluation purposes.

To establish benchmarks (evaluation frameworks) for music scene description by labelling copyright-cleared musical pieces with correct descriptions, a multipurpose music-scene labelling editor (metadata editor) was also developed [225]. It enables a user to hand-label a musical piece with music scene descriptions shown in Fig. 11.1. The editor can deal with both audio files and standard MIDI files and supports interactive audio/MIDI playback while editing. Along a wave or MIDI piano-roll display it shows subwindows in which any selected descriptions can be displayed and edited. To facilitate the support of various descriptions, its architecture is based on a plug-in system in which an external module for editing each description is installed as plug-in software. As a first step, the RefraiD method was evaluated by using the chorus section metadata for 100 songs of the RWC Music Database: Popular Music (80 of the 100 songs were correctly detected) [224].

11.7 Applications of Music Scene Description

Music scene description methods that can deal with real-world audio signals of musical pieces sampled from CD recordings have various practical applications such as music information retrieval, music-synchronized computer graphics, and music listening stations. The following sections introduce these applications.

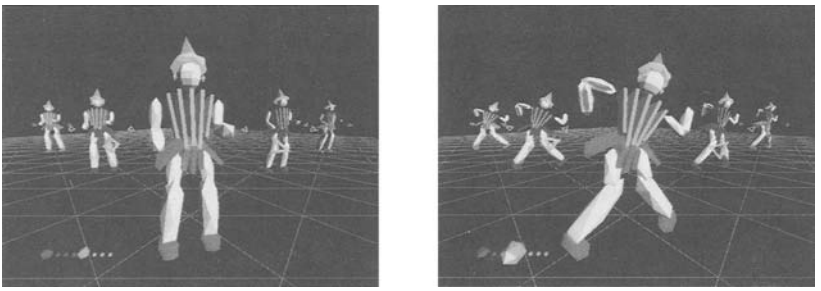


Fig. 11.13. Virtual dancer 'Cindy'.

11.7.1 Music Information Retrieval

Music scene description contributes to content-based music information retrieval since it can provide various acoustical metadata (annotations) of musical pieces. For example, the automatic melody estimation described in Section 11.2 is useful for *query by humming (QBH)* [323], [207], [604], [484], [603], [507], [605], [585], [301], [109] which enables a user to retrieve a musical piece by humming or singing its melody: a QBH database consisting of audio signals of musical pieces can be indexed using their melody lines. Moreover, the description of chorus sections (Section 11.5) can increase the efficiency and precision of QBH by enabling a QBH system to match a query with only the chorus sections.

Temporal or rhythmic descriptions such as beat structure, tempo, and drums (Sections 11.3 and 11.4) are also useful for retrieving musical pieces on the basis of rhythm and tempo. Indexing musical pieces using drum descriptions, for example, will enable a user to retrieve music by voice percussion or beat boxing (verbalized expression of drum sounds by voice) [479], [326].

In addition, various music scene descriptions facilitate the computation of similarity between musical pieces. Similarity measures based on music scene descriptions enable a user to use musical pieces themselves as the search key to retrieve a musical piece having a similar feeling. These measures can also be used to automatically classify musical pieces into genres or music styles.

11.7.2 Music-Synchronized Computer Graphics

Because the beat tracking described in Section 11.3 and Chapter 4 can be used to automate the time-consuming tasks that must be done to synchronize events with music, there are various applications. In fact, Goto and Muraoka [235], [220], [221] developed a real-time system that displays virtual dancers and several graphic objects whose motions and positions change in time to beats (Fig. 11.13). This system has several dance sequences, each for a different mood of dance motions. While a user selects a dance sequence manually, the timing of each motion in the selected sequence is determined automatically

on the basis of the beat-tracking results. Such a computer graphics system is suitable for live stage, TV program, and karaoke uses.

Beat tracking also facilitates the automatic synchronization of computer-controlled stage lighting with the beats in a musical performance. Various properties of lighting—such as colour, brightness, and direction—can be changed in time to music. In the above virtual dancer system, this was simulated on a computer graphics display with virtual dancers.

11.7.3 Music Listening Station

The automatic chorus section detection described in Section 11.5 enables new music-playback interfaces that facilitate content-based manual browsing of entire songs. As an application of the RefraiD method, Goto [226] developed a music listening station for trial listening, called SmartMusicKIOSK. Customers in music stores often search out the chorus or 'hook' of a song by repeatedly pressing the fast-forward button, rather than passively listening to the music. This activity is not well supported by current technology. SmartMusicKIOSK provides the following two functions to facilitate an active listening experience by eliminating the hassle of manually searching for the chorus and making it easier for a listener to find desired parts of a song:

1. *'Jump to chorus' function: automatic jumping to the beginning of sections relevant to a song's structure*
   Functions are provided enabling automatic jumping to sections that will be of interest to listeners. These functions are 'jump to chorus (NEXT CHORUS button)', 'jump to previous section in song (PREV SECTION button)', and 'jump to next section in song (NEXT SECTION button)', and they can be invoked by pushing the buttons shown above in parentheses (in the lower window of Fig. 11.14). With these functions, a listener can directly jump to and listen to chorus sections, or jump to the previous or next repeated section of the song.
2. *'Music map' function: visualization of song contents*
   A function is provided to enable the contents of a song to be visualized to help the listener decide where to jump next. Specifically, this function provides a visual representation of the song's structure consisting of chorus sections and repeated sections, as shown in the upper window of Fig. 11.14. While examining this display, the listener can use the automatic jump buttons, the usual fast-forward/rewind buttons, or a playback slider to move to any point of interest in the song.

This interface, which enables a listener to look for a section of interest by interactively changing the playback position, is useful not only for trial listening but also for more general purposes in selecting and using music. While entire songs of no interest to a listener can be skipped on conventional music-playback interfaces, SmartMusicKIOSK is the first interface that allows the listener to easily skip sections of no interest even within a song.

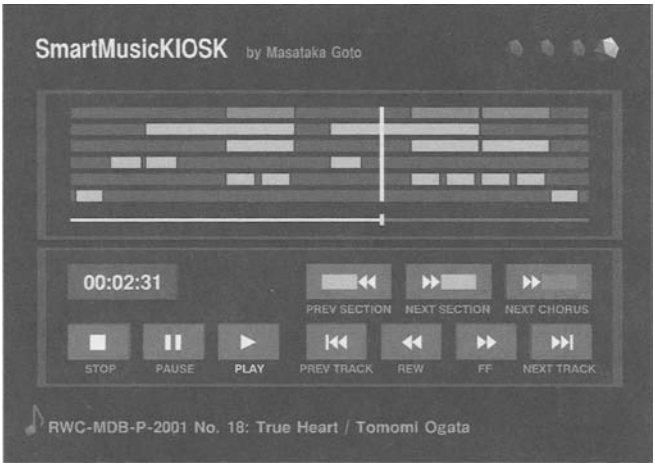


Fig. 11.14. SmartMusicKIOSK screen display. The lower window provides content-based controls allowing a listener to skim rapidly through music as well as common playback controls. The upper window provides a graphical overview of the music structure (results of automatic chorus section detection using RWC-MDB-P-2001 No. 18 of the RWC Music Database [229], [227]). The horizontal axis of the upper window is the time axis covering the entire song; the top row shows chorus sections, the five lower rows show repeated sections, and the bottom horizontal bar is a playback slider.

11.8 Conclusion

This chapter has described the music scene description research approach towards developing a system that understands real-world musical audio signals without deriving musical scores or separating signals. This approach is important from an academic viewpoint because it explores what is essential for understanding audio signals in a human-like fashion. The ideas and techniques are expected to be extended to not only music signals but also general audio signals including music, speech, environmental sounds, and mixtures of them. Traditional speech recognition frameworks have been developed for dealing with only monophonic speech signals or a single-pitch sound with background noise, which should be removed or suppressed without considering their relationship. Research on understanding musical audio signals is a good starting point for creating a new framework for understanding general audio signals, because music is polyphonic, temporally structured, and complex, yet still well organized. In particular, relationships between various simultaneous or successive sounds are important and unique to music. This chapter, as well as other chapters in this book, will contribute to such a general framework.

The music scene description approach is also important from industrial or application viewpoints since end users can now easily 'rip' audio signals from CDs, compress and store them on a personal computer, load a huge number of

songs onto a portable music player, and listen to them anywhere and anytime. These users want to retrieve and listen to their favourite music or a portion of a musical piece in a convenient and flexible way. Reflecting these demands, the target of processing has expanded from the internal content of individual musical pieces to entire musical pieces and even sets of musical pieces [233]. While the primary target of music scene description is the internal content of a piece, the obtained descriptions are useful for dealing with sets of musical pieces as described in Section 11.7.1. The more accurate and detailed we can make the obtained music scene descriptions, the more advanced and intelligent music applications and interfaces will become.

Although various methods for detecting melody and bass lines, tracking beats, detecting drums, and finding chorus sections have been developed and successful results have been achieved to some extent, there is much room for improving these methods and developing new ones. For example, in general each method has been researched independently and implemented separately. An integrated method exploiting the relationships between these descriptions will be a promising next step. Other music scene descriptions apart from those described in this chapter should also be investigated in the future. Ten years ago it was considered too difficult for a computer to obtain most of the music scene descriptions described here, but today we can obtain them with a certain accuracy. I look forward to experiencing further advances in the next ten years.