Specmurt Analysis of Polyphonic Music Signals

Shoichiro Saito, *Student Member, IEEE*, Hirokazu Kameoka, *Student Member, IEEE*, Keigo Takahashi, Takuya Nishimoto, *Member, IEEE*, and Shigeki Sagayama, *Member, IEEE*

***Abstract*—This paper introduces a new music signal processing method to extract multiple fundamental frequencies, which we call *specmurt* analysis. In contrast with *cepstrum* which is the inverse Fourier transform of log-scaled power spectrum with linear frequency, specmurt is defined as the inverse Fourier transform of linear power spectrum with log-scaled frequency. Assuming that all tones in a polyphonic sound have a common harmonic pattern, the sound spectrum can be regarded as a sum of linearly stretched common harmonic structures along frequency. In the log-frequency domain, it is formulated as the convolution of a common harmonic structure and the distribution density of the fundamental frequencies of multiple tones. The fundamental frequency distribution can be found by deconvolving the observed spectrum with the assumed common harmonic structure, where the common harmonic structure is given heuristically or quasi-optimized with an iterative algorithm. The efficiency of specmurt analysis is experimentally demonstrated through generation of a piano-roll-like display from a polyphonic music signal and automatic sound-to-MIDI conversion. Multipitch estimation accuracy is evaluated over several polyphonic music signals and compared with manually annotated MIDI data.**

***Index Terms*—Inverse filtering, iteration algorithm, multipitch analysis, pitch visualization, polyphonic music signals.**

I. Introduction

In 1963, Bogert, Healy, and Tukey introduced the concept of "cepstrum" in a paper entitled "*The quefrency alanysis of time series for echoes: cepstrum, pseudoautocovariance, cross-cepstrum, and saphe-cracking*" [1] where they defined cepstrum as the inverse Fourier transform of logarithmically scaled power spectrum. Their humorous terminologies such as "quefrency" and "lifter" which are anagrams of "frequency" and "filter," respectively, have been since widely used in the speech recognition area.

Manuscript received February 26, 2007; revised September 21, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hong-Goo Kang.

S. Saito was with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan. He is now with NTT Cyber Space Laboratories, Tokyo 180-8585, Japan (e-mail: saito@hil.t.u-tokyo.ac.jp).

H. Kameoka was with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan. He is now with NTT Communication Science Laboratories, Atsugi 243-0198, Japan (e-mail: kameoka@hil.t.u-tokyo.ac.jp).

K. Takahashi was with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan. He is now with the Community Safety Bureau, National Police Agency, Tokyo 100-8974, Japan (e-mail: takahashi@hil.t.u-tokyo.ac.jp).

T. Nishimoto and S. Sagayama are with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (e-mail: nishi@hil.t.u-tokyo.ac.jp; sagayama@hil.t.u-tokyo.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2007.912998

Since Noll [2] used cepstrum in pitch detection in 1964, it became a standard technique for detection and extraction of fundamental frequency of periodic signals. Later, cepstrum became a major feature parameter for speech recognition in the late 1970s together with delta-cepstrum [3] and "Mel-frequency cepstrum coefficients" (MFCCs) [4]. Cepstrum was also used as filter coefficients in speech synthesis digital filter [5] and plays a central role in HMM-based speech synthesis.

In these applications, cepstrum is advantageous as it converts the speech spectrum into the sum of spectral fine structure (pitch information) and spectral envelope components in the cepstrum domain. It is usually assumed, however, that the target is a single pitch (or, one speaker's voice) signal, and multipitch signals cannot be well handled by the cepstrum due to the nonlinearity of the logarithm.

Multipitch analysis has been one of the major concerns in music signal processing. It has a wide range of potential applications including automatic music transcription, score following, melody extraction, automatic accompaniment, music indexing for music information retrieval, etc. However, fundamental frequency cannot be easily detected from a multipitch audio signal, i.e., polyphonic music, due to spectral overlap of overtones, poor frequency resolution, spectral widening in short-time analysis, etc. Various approaches concerning the multipitch detection/estimation problem have been attempted since the 1970s as extensively described in [6]. In the mid 1990s, approaches combining artificial intelligence and computational auditory scene analysis with signal processing were considered (see, for example, [7]). In recent years, more analytical approaches have been investigated, aiming at a higher accuracy. In one of the earliest attempts in this direction, Brown [8] considered harmonic pattern on the logarithmic frequency axis and used convolution to calculate the cross correlation with a reference pattern, expecting a major peak at the fundamental frequency. This idea is essentially a "matched filter" in the log-frequency domain, and it can be put in contrast with the method presented in this paper as explained in Section III-F. Other approaches include the combination of a probabilistic approach with multiagent systems for predominant-F0 estimation [9]–[11], nonnegative matrix factorization [12], [13], sparse coding in frequency domain [14] or time domain [15], Gaussian harmonic models [16], linear models for the overtone series [17], harmonicity and spectral smoothness [18], harmonic clustering [19], and use of information criterion for the estimation of the number of sound sources [20].

As for spectral analysis, wavelet transform using the Gabor function is one of the popular approaches to derive short-time power spectrum of music signals along the logarithmically scaled frequency axis, which appropriately suits the music pitch scaling. Spectrogram, i.e., the 2-D time–frequency display of the sequence of short-time spectra, however, can look very intricate because of the existence of many overtones (i.e.,

the harmonic components of multiple fundamental frequencies), that often prevents us from discovering music notes.

This paper introduces *specmurt* analysis, a technique based on the Fourier transform of logarithmically transformed power spectrum, which is effective for multipitch analysis of polyphonic music signals. Our objective is to emphasize the fundamental frequency components by suppressing the harmonic components on the spectrogram. The obtained spectrogram then becomes more similar to a piano-roll display from which multiple fundamental frequencies can be easily identified. The approach of the proposed method entirely differs from that of the standard multipitch analysis methods that determine uniquely the most likely solutions to the multipitch detection/estimation problem. In many of these methods, the number of sources needs to be decided before the methods are applied, but specmurt analysis does not require such a decision, and the output result contains information about the number of sources. Specmurt analysis provides a display which is visually similar to the original piano-roll image and shall hopefully be a useful feature, for example, for retrieval purposes (one could, for instance, imagine a simple image template matching).

The overview of this paper is as follows: in Section II, we discuss the relationship between cepstrum and specmurt. In Section III, we introduce a multipitch analysis algorithm using specmurt. Furthermore, we describe an algorithm for iterative estimation of the common harmonic structure in Section IV and in the Appendix , and finally we show experimental results of multipitch estimation, followed by discussion and conclusion.

II. "Cepstrum" Versus "Specmurt"

A. Cepstrum

According to the Wiener–Khinchin theorem, the inverse Fourier transform of the linear power spectrum with linear frequency is the autocorrelation as a function of time delay

$$v(\tau) = \int_{-\infty}^{\infty} f(\omega) e^{j\tau\omega} d\omega, \quad -\infty < \tau < \infty \tag{1}$$

where $f(\omega)$ denotes the power spectrum of the signal. If the power spectrum is scaled logarithmically, the resulting inverse Fourier transform is not the autocorrelation any more and has been named 'cepstrum' [1], humorously reversing the first four letters of 'spectrum'. It is defined as follows:

$$c(q) = \int_{-\infty}^{\infty} \log f(\omega) e^{jq\omega} d\omega, \quad -\infty < q < \infty \tag{2}$$

where q is called "quefrency." This transform has become an important tool in speech recognition.

Cepstrum is one of the standard methods for finding a single fundamental frequency. However, multiple fundamental frequencies cannot be handled appropriately since, after the nonlinear scaling procedure, the spectrum is no longer a linear combination of sources, even in the expectation sense.

B. Specmurt

Instead of inverse Fourier transform of log-scaled power spectrum with linear frequency , we can alternatively consider inverse Fourier transform of linear power spectrum with log-scaled frequency as follows:

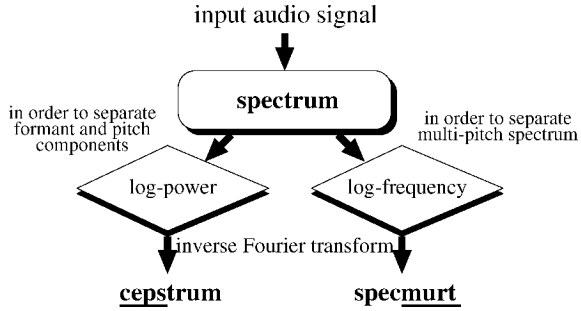


Fig. 1. Comparison between cepstrum and specmurt: specmurt is defined as the inverse Fourier transform of the linear spectrum with log-frequency, whereas cepstrum is the inverse Fourier transform of the log spectrum with linear frequency.

$$s(y) = \int_{-\infty}^{\infty} f(\omega) e^{jy \log|\omega|} d\log|\omega|, \quad -\infty < y < \infty \tag{3}$$

or, denoting $x = \log|\omega|$ and $g(x) = f(\omega) + f(-\omega)$:

$$s(y) = \int_{-\infty}^{\infty} g(x) e^{jxy} dx, \quad -\infty < y < \infty \tag{4}$$

which we call specmurt by reversing the last four letters in the spelling of "spectrum," by analogy with the terminology of cepstrum where the first four letters of "spectrum" are reversed (see Fig. 1).

In the following section, we will show that specmurt is effective in multipitch signal analysis, while cepstrum can be used for the single-pitch case.

It should be noted that the above definition can be rewritten as a special case of the Mellin transform on the imaginary axis

$$s(y) = \int_{-\infty}^{\infty} \omega^{jy-1} f(\omega) d\omega, \quad -\infty < y < \infty\,. \tag{5}$$

However, we still use the terminology "specmurt" to emphasize its relationship with cepstrum and to avoid confusion with the Mellin transform on the real axis, which is widely used to derive scale-invariant features [21]. Obviously, specmurt preserves the scale, and is thus useful in finding multiple fundamental frequencies as we shall show in later sections. In addition, we will need to make use of the convolution theorem of the Fourier transform to deconvolve the harmonic structure, but this theorem is missing from the basic properties of the Mellin transform.

It should be emphasized again that specmurt uses a linear scale for the power of the spectrum, in comparison with MFCCs which are very often used in feature analysis in speech recognition. Moreover, when logarithmically scaled both in frequency and magnitude, the spectrum is called Bode diagram, which is often used in automatic control theory, and the Mel-generalized cepstral analysis as proposed in [22].

Practically, spectrum analysis with logarithmic scale is performed using (continuous) wavelet transform

$$W(a,b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}^{*}(t) \mathrm{d}t \tag{6}$$

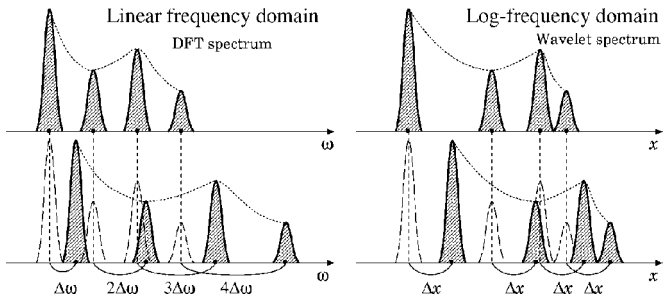


Fig. 2. Relative location of fundamental frequency and harmonic frequencies both in linear and log scale.

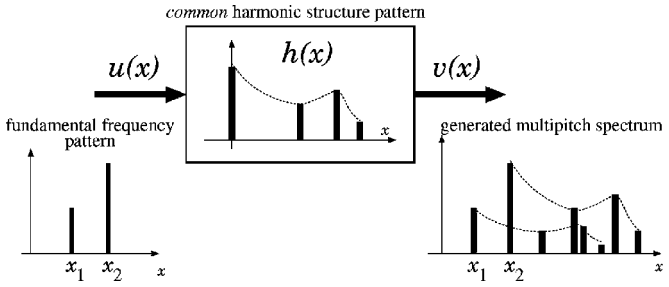


Fig. 3. Multipitch spectrum generated by convolution of a fundamental frequency pattern and a common harmonic structure pattern.

where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right). \tag{7}$$

$x(t)$ denotes the target signal, $\psi(t)$ is the mother wavelet, and ψ^* is the complex conjugate of ψ. In this paper, Gabor function is used as the mother wavelet

$$\psi(t) = \frac{1}{\sqrt{2\pi}\sigma}e^{-t^2/2\sigma^2 + j\omega_0 t} \tag{8}$$

so as to obtain a short-time power spectrum with a constant resolution along the log-frequency axis. It can be understood as constant-Q filter bank analysis along the log-scaled frequency axis and is well suited for the musical pitch scale.

III. Specmurt Analysis of Multipitch Spectrum

A. Modeling Single-Pitch Spectrum in Log-Frequency Domain

Assuming that a single sound component is a harmonic signal, the frequencies of the second, third, etc. harmonics are integer multiples of the fundamental frequency in linear frequency scale. This means that if the fundamental frequency changes by $\Delta\omega$, the nth harmonic frequency changes by $n\Delta\omega$. In the logarithmic frequency (log-frequency) scale, on the other hand, the harmonic frequencies are located at $\log\omega + \log 2,\ \log\omega + \log 3,\ \ldots,\ \log\omega + \log n$, where $\log\omega$ is the fundamental log-frequency. The relative location thus remains constant no matter how the fundamental frequency changes and undergoes an overall parallel shift depending on the change (see Fig. 2).

Nothing is new in the above discussion: music pitch interval can be described using semitones, which is equivalent to log-frequency. This relation has been explicitly or implicitly used for multipitch analysis, for example in [8] and [9].

B. Common Harmonic Structure

Let us define here a general spectral pattern for a single harmonic sound. The assumption that the relative powers of its harmonic components are common and do not depend on its fundamental frequency suggests a general model of harmonic structure. We call this pattern the *common harmonic structure* and denote it as $h(x)$, where x indicates log-frequency. The fundamental frequency position of this pattern is set to the origin (see Fig. 3). Under this definition, we can explicitly obtain the spectrum of a single harmonic sound by convolving an impulse function (Dirac's delta-function) $u_\delta(x)$ and the common harmonic structure $h(x)$. Here the position of the impulse represents the fundamental frequency of the single sound on the x-axis and the height represents the energy.

In reality, the harmonic structure varies with the fundamental frequency even for a given musical instrument. However, the purpose of this assumption is not to model the spectrum of music signals strictly, and the result includes the modeling error by definition. Nevertheless, this strong assumption enables us to reach a simple, quick, and acceptably accurate solution.

C. Modeling Multipitch Spectrum in Log-Frequency Domain

If $u(x)$ contains power at multiple fundamental frequencies as shown in Fig. 3, the multipitch spectrum $v(x)$ is generated by convolution of $h(x)$ and $u(x)$

$$v(x) = h(x) * u(x) \tag{9}$$

if the power spectrum can be assumed additive ($*$ denotes convolution). Actually, when summing up multiple sinusoids at the same frequency, the power of the signal may deviate from the sum of each sinusoidal powers due to their relative phase relationship. However, this assumption holds in the expectation sense.

Note that (9) still holds if $u(x)$ consists not of multiple delta functions but of a continuous function representing the distribution of fundamental frequencies.

D. Deconvolution of Log-Frequency Spectrum

The main objective here is to estimate the fundamental frequency pattern $u(x)$ from the observed spectrum $v(x)$. If the common harmonic structure $h(x)$ is known, we can recover $u(x)$ by applying the inverse filter $h^{-1}(x)$ to $v(x)$. It corresponds to the deconvolution of the observed spectrum $v(x)$ by the common harmonic structure pattern $h(x)$

$$u(x) = h^{-1}(x) * v(x). \tag{10}$$

In the Fourier domain, this equation can be easily computed by division of the inverse Fourier transform of the log-frequency linear-amplitude power spectrum by the inverse Fourier transform of the common harmonic structure

$$U(y) = \frac{V(y)}{H(y)} \tag{11}$$

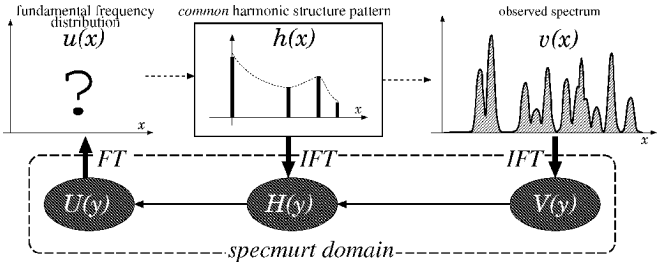


Fig. 4. Outline of multiple fundamental frequency estimation through specmurt analysis. Fundamental frequency distribution $u(x)$ is calculated through the division $V(y)/H(y)$.

where $U(y)$, $H(y)$, and $V(y)$ are the inverse Fourier transform of $u(x)$, $h(x)$, and $v(x)$, respectively. The fundamental frequency pattern $u(x)$ is then restored by

$$u(x) = \mathcal{F}[\, U(y) \,]\,. \tag{12}$$

The y domain has been defined as the inverse Fourier transform of linear spectrum magnitude with logarithmic frequency x and it is equivalent to specmurt domain, as mentioned in Section II-B. We call this procedure specmurt analysis. In practical use, it is indifferent whether the definition of the y domain is inverse Fourier transform or Fourier transform of the x domain, and here we choose the former definition in contrast with cepstrum definition.

E. Computational Procedure of Specmurt Analysis

The whole procedure of specmurt analysis consists of four steps as shown below.

1) Apply wavelet transform with Gabor function to the input signal and take the squared absolute values (power-spectrogram magnitudes) $v(x)$ for each frame.
2) Apply inverse Fourier transform to $v(x)$ to obtain $V(y)$.
3) Divide $V(y)$ by $H(y)$, the inverse Fourier transform of the assumed common harmonic pattern $h(x)$.
4) Fourier transform the division $V(y)/H(y)$ to estimate the multipitch distribution $u(x)$ along the log-frequency x.

The term "frame" in this paper means a certain discrete time shift parameter, denoted by b in (6), not the short time interval of the signals. Wavelet transform does not utilize the short time frame, but the obtained spectra for each time shift parameter b can be treated almost the same as the spectra obtained by the short time Fourier transform. For this reason, we call the discrete time shift in wavelet transform "frame" in this paper.

This process is briefly illustrated in Fig. 4. The process is done over every short-time analysis frame and thus we finally have a time series of fundamental frequency components, i.e., a piano-roll-like visual representation with a small amount of computation.

The discussion has been conducted so far under the assumption that the common harmonic structure pattern is common over all constituent tones and also known *a priori*. Even in actual situations where this assumption may not strictly hold, this approach is still expected to play an effective role as a fundamental frequency component emphasis (or, in other words, overtone suppression).

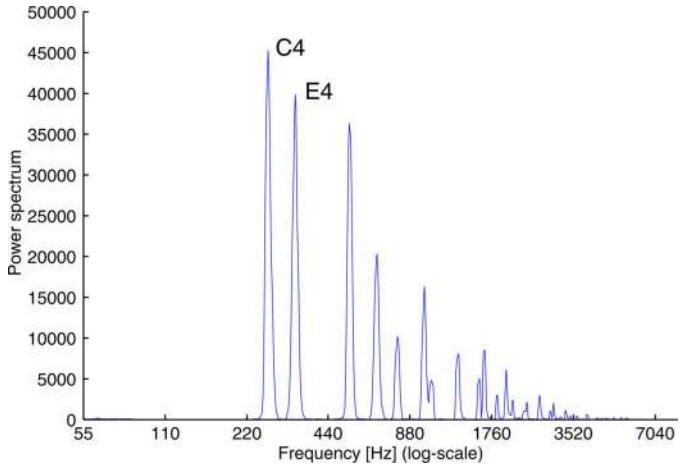


Fig. 5. Wavelet transform of two mixed violin sounds (C4 and E4).

F. Inverse Filter Versus Matched Filter

Using logarithmic frequency is a common idea in music where pitch is perceived logarithmically. Brown [8] actually attempted to emphasize the fundamental frequency by *convolution* of the spectrum with a reference harmonic pattern on the log-frequency axis to calculate the cross-correlation, whereas we aim at emphasizing the fundamental frequency by *deconvolution* of the spectrum by a common harmonic pattern. The former is a "matched filter" approach while the latter is an "inverse filter" approach from the filter theory.

In single pitch estimation of speech, autocorrelation of the prediction residuals obtained by inverse filtering of speech signals with linear predictive coefficients (LPCs) [23], [24] is more effective to estimate precisely the pitch frequency than simple autocorrelation of the signals.

IV. Quasi-Optimization of the Common Harmonic Structure

In the procedure described above to perform specmurt analysis, we assumed that all constituent sounds have a common harmonic structure. It is, however, generally not true in real polyphonic music sounds as the harmonic structures are generally different from each other, and often change over time. The variation of the harmonic structure between sounds inside a frame is not considered in specmurt, as it is modeled as a linear system, but concerning the variation in time, there is still room to adapt the harmonic structure to the quasi-optimal pattern frame by frame (the term "quasi-optimal" means that the result converges after iteration of the algorithm but the effective function of the whole algorithm measuring the optimality is not defined). The best we can do is to estimate $h(x)$ such that it minimizes the amplitudes of overtones in $u(x)$ after deconvolution.

Fig. 5 shows as an example the linear-scaled spectrum of a mixture of two audio violin sounds (C4 and E4, excerpted from RWC Musical Instrument Sound Database [25]) along log-scaled frequency axis x, where the multiple peaks represent the two fundamental frequencies as well as the overtones. If we use $1/\sqrt{f}$ as the frequency characteristic of $h(x)$, where f denotes frequency (shown in Fig. 6(I-a)), the overtones are attenuated

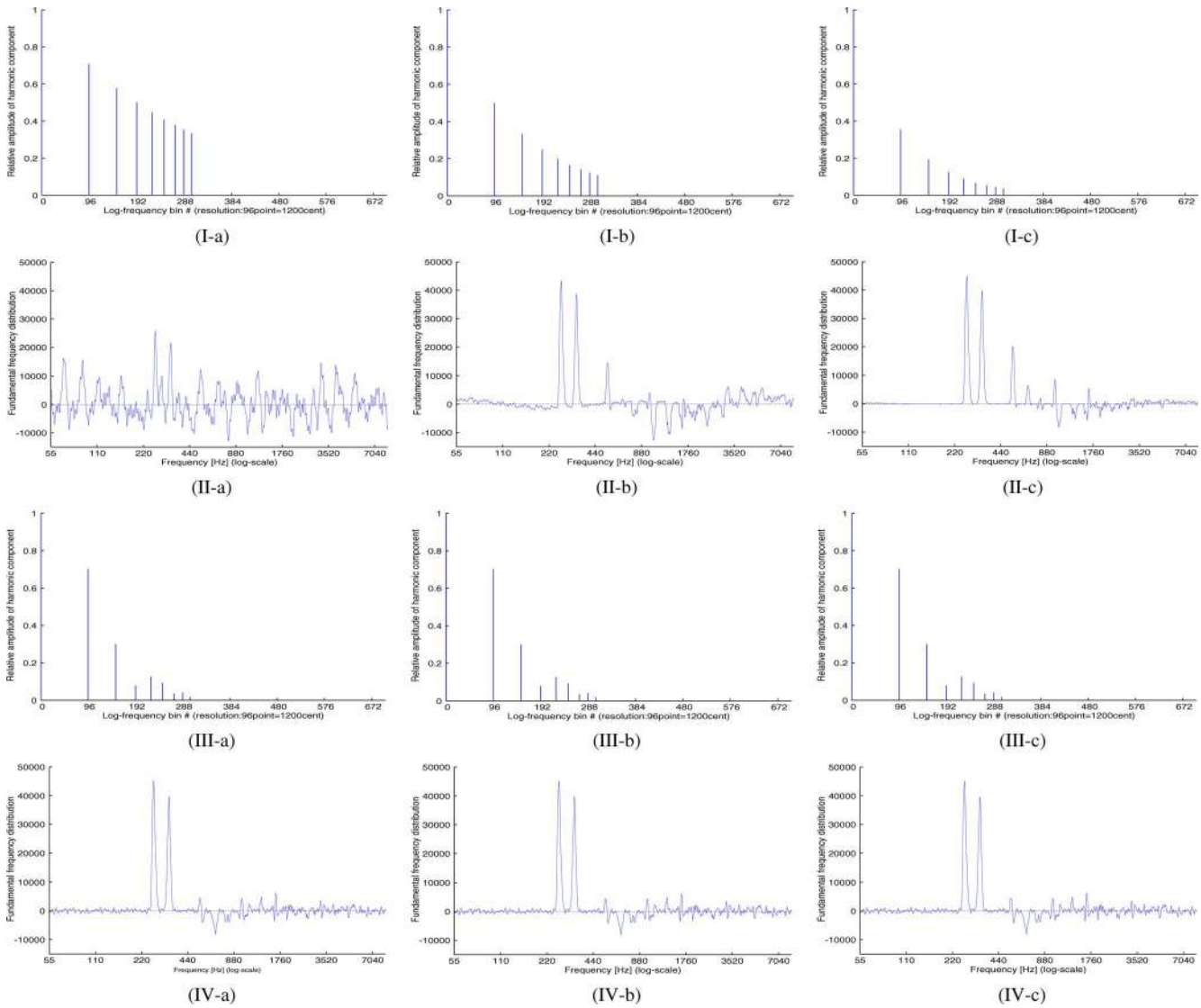


Fig. 6. Overtone suppression results for the spectrum of Fig. 5 with three different initial harmonic structures (a,b,c). (I) Initial value of the common harmonic structure (from left to right, the harmonic structure envelope is $1/\sqrt{f}$, $1/f$, $1/f^{1.5}$, respectively). (II) Fundamental frequency distribution before performing any iteration. (III) Estimated common harmonic structure after five iterations. (IV) Improved fundamental frequency distribution after five iterations. The three estimations with different initial value converge to almost the same result.

but the power is strongly fluctuating and many unwanted components in the entire range of frequency appear as the result of deconvolution (Fig. 6(II-a)). On the other hand, if we use $1/f$ or $1/f^{1.5}$ (Fig. 6(I-b) and (I-c), respectively), overtone suppression is insufficient (Fig. 6(II-b) and (II-c)). In this case the result of Fig. 6(II-b) seems to be the best of the three, but in general it is unrealistic to find out manually an appropriate harmonic structure at every analysis frame.

Hence, it is desirable to estimate automatically the quasi-optimal $h(x)$ that gives maximum suppression of overtone components. However, specmurt analysis is an "inverse filtering" process and it is an ill-posed problem when both the fundamental frequency distribution $u(x)$ and the common harmonic structure $h(x)$ are completely unknown. In other words, we need to impose some constraints on the solution set in order to select an appropriate solution from an infinitely large number of choices. The following describes an iterative estimation algorithm that utilizes two constraints on $u(x)$ and $h(x)$ and calculates a quasi-optimal solution.

A. Nonlinear Mapping of the Fundamental Frequency Distribution

Here, we introduce the first constraint: the fundamental frequency distribution is nearly zero almost everywhere, except for some predominant peaks. In other words, the fundamental frequency distribution is sparse. This means that the minor peaks of $u(x)$ are not the real fundamental frequency components but errors in the specmurt analysis. It is difficult, however, to distinguish with certainty between the real fundamental frequency components and the unwanted ones, because of the variety of relationships between the peak amplitudes of both types.

In consideration of this problem, we introduce a nonlinear mapping function to update the fundamental frequency distribu-

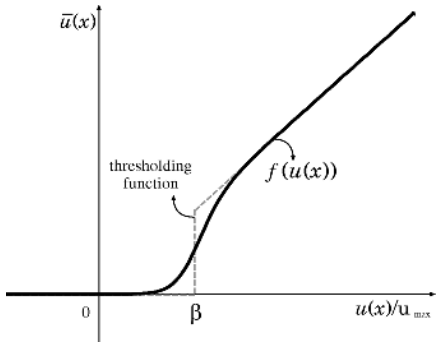


Fig. 7. Nonlinear mapping function provides fuzziness and does not suppress completely the lower value than β. Solid line: nonlinear mapping function to suppress minor peaks and negative values of $u(x)$; dashed line: hard thresholding function.

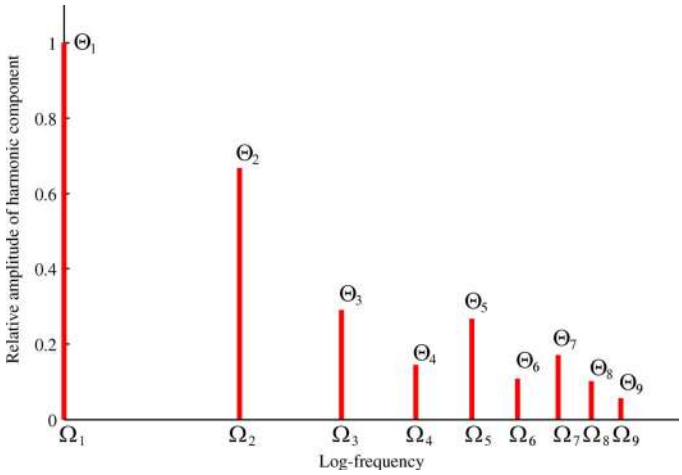


Fig. 8. Illustration of the parameterized common harmonic structure $\bar{h}(x,\Theta)$. Ω_n is the location of the nth harmonic component in log-frequency scale, and Θ_n is the nth relative amplitude. $\Theta_2, \Theta_3, \ldots, \Theta_N$ are variable and should be estimated ($\Theta_1 = 1$).

tion, which avoids having to make a hard decision and provides fuzziness. It is defined as follows:

$$\bar{u}(x) = \frac{1}{1+\exp\left\{-\alpha\left(\frac{u(x)}{u_{\max}}-\beta\right)\right\}}u(x) \tag{13}$$

where $u_{\max}$ stands for $\max\{u(x), \forall x\}$. It is shown in Fig. 7. This mapping uses a sigmoid function and has a fuzziness parameter α and a threshold magnitude parameter β. β corresponds to the value under which frequency components are assumed to be unwanted, and α represents the degree of fuzziness of the boundary ($\alpha > 0$).

This nonlinear mapping does not change the values which are significantly larger than β, and attenuates both the slightly larger and the smaller values. The degree of attenuation becomes stronger as the value concerned is small. The hard thresholding function is also shown in Fig. 7 as a dashed line. Compared with the nonlinear mapping, it does not change the values which are larger than β, and sets the smaller values to zero

$$\bar{u}(x) = \begin{cases} u(x) & \left(if\ \frac{u(x)}{u_{\max}} \geq \beta\right) \\ 0 & \left(if\ \frac{u(x)}{u_{\max}} < \beta\right) \end{cases}. \tag{14}$$

The nonlinear mapping function depends less arbitrarily on β: when the hard thresholding function is applied to values around β, $\bar{u}(x)$ can result in a totally different value for a small change of β. In contrast, the nonlinear mapping does not have a abrupt threshold under which the values are set to zero, instead, the change occurs more gradually. Therefore, it does not suffer from this problem, and a small change in parameter α does not influence drastically the value of $\bar{u}(x)$. Consequently, we do not have to make a strict decision on the threshold of the amplitude between the fundamental frequency components and the other ones. In fact, the nonlinear mapping is a broader concept than thresholding, as the nonlinear mapping with $\alpha = +\infty$ actually corresponds to the hard thresholding.

Although the nonlinear mapping does not change $u(x)$ widely, after a few iterations $u(x)$ becomes sparse enough. This mapping decreases the value of $u(x)$ for all x, but if $v(x_0)$ has a certain amount of amplitude and x_0 does not correspond to a harmonics frequency, $u(x_0)$ can increase back from the attenuated value at the deconvolution step (an example is shown in Section IV-C).

As a result of the mapping, the components of $u(x)$ with small or negative power are brought close to zero, while middle power components remain as slightly smaller peaks. This means that $\bar{u}(x)$ should be closer to the ideal fundamental frequency distribution than $u(x)$, as the small nonlikely peaks have been reduced.

B. Common Harmonic Structure Estimation

In the previous section, we introduced $\bar{u}(x)$ as a more preferable distribution than $u(x)$, and we can now calculate the most suitable common harmonic structure from $\bar{u}(x)$ and the observed spectrum $v(x)$. We shall consider here a second constraint about the common harmonic structure $h(x)$: a common harmonic structure is composed of a certain number of impulse components located at the positions of the harmonics in log-scale. More precisely

$$\bar{h}(x,\boldsymbol{\Theta}) = \sum_{n=1}^{N} \Theta_n \delta(x-\Omega_n) \tag{15}$$

where Ω_n and Θ_n are, respectively, the x-coordinate and the relative amplitude of the nth harmonic overtone in log-frequency scale, N is the number of harmonics to consider ($\Omega_1 = 0$ and $\Theta_1 = 1$), and $\Omega_n = \log n \cdot 1200/r$ (the overview of $h(x,\boldsymbol{\Theta})$ is illustrated in Fig. 8). $r[\mathrm{cent/bin}]$ is the (log-)frequency resolution of the wavelet transform. Under this constraint, we calculate the common harmonic structure by estimating the parameter Θ, which is done through minimization of the square error

$$E(\boldsymbol{\Theta}) = \int_{-\infty}^{\infty} \left\{v(x) - \bar{h}(x,\boldsymbol{\Theta}) * \bar{u}(x)\right\}^2 dx. \tag{16}$$

This objective function is quadratic in the parameters Θ_n and the quasi-optimal solution can be obtained by considering $N-1$ partial differential equations

$$\frac{\partial E(\boldsymbol{\Theta})}{\partial \Theta_n} = 0 \ \ (\forall n,\ 2 \leqq n \leqq N) \tag{17}$$

or, in detail

$$\begin{pmatrix} a_{2,2} & \cdots & a_{2,n} & \cdots & a_{2,N} \\ \vdots & & \vdots & & \vdots \\ a_{n,2} & \cdots & a_{n,n} & \cdots & a_{n,N} \\ \vdots & & \vdots & & \vdots \\ a_{N,2} & \cdots & a_{N,n} & \cdots & a_{N,N} \end{pmatrix} \begin{pmatrix} \Theta_2 \\ \vdots \\ \Theta_n \\ \vdots \\ \Theta_N \end{pmatrix} = \begin{pmatrix} b_2 \\ \vdots \\ b_n \\ \vdots \\ b_N \end{pmatrix} \tag{18}$$

where

$$a_{j,k} = \int_{-\infty}^{\infty} \bar{u}(x-\Omega_j)\bar{u}(x-\Omega_k)dx, \tag{19}$$

$$b_j = \int_{-\infty}^{\infty} \{v(x)-\bar{u}(x)\}\,\bar{u}(x-\Omega_j)dx. \tag{20}$$

The optimal parameter Θ can then be obtained by solving (18), which can be done because the non-singularity of the matrix involved is guaranteed, as proved in the Appendix .

We can now use again the specmurt analysis procedure to obtain a yet improved $u(x)$ using the improved common harmonic structure $\bar{h}(x)$.

C. Iterative Estimation Algorithm

Practically, the quasi-optimal harmonic structure is obtained by iterating the above procedures. Summarizing the above, the iterative algorithm goes as follows.

- Step 1) Obtain $u(x)$ from $v(x)$ with initial $h(x)$ by inverse filtering.
- Step 2) Obtain $\bar{u}(x)$ by applying a nonlinear mapping.
- Step 3) Find $\bar{h}(x,\boldsymbol{\Theta})$ at $N-1$ discrete points $\{\Omega_2,\Omega_3,\ldots,\Omega_N\}$ by calculating $\{\Theta_2,\Theta_3,\ldots,\Theta_N\}$.
- Step 4) Replace $h(x)$ with $\bar{h}(x,\boldsymbol{\Theta})$ and go back to Step 1).

In Step 2), all the spectral components are attenuated according to their amplitudes, but fundamental frequency components get back their original amplitude in the next Step 1) (see the experiment in Section IV-D). Although the convergence of this procedure for optimizing the common harmonic structure is not mathematically guaranteed, we have not experienced any serious problem in this matter. In addition, we also considered a probabilistic model and applied it to specmurt analysis in another paper [26]. In that algorithm, the convergence is guaranteed but at the expense of a slightly more complicated formulation.

D. Implementation and Examples

In order to implement this algorithm, we need to translate the above discussion from continuous analysis to discrete analysis to enable the computational calculation. The integral calculation is approximated by summation at finite range, and log-scaled location of harmonics component Ω_n is rounded to nearest frequency bin.

An example illustrating the iterative quasi-optimization is shown in Fig. 6(III)–(IV). The above procedure is performed starting from three types of initial $h(x)$ in Fig. 6(I-a)–(I-c). The quasi-optimized common harmonic structures after five iterations are shown in Fig. 6(III-a)–(III-c) and the corresponding fundamental frequency distributions are shown in Fig. 6(IV-a)–(IV-c). In this experiment, the parameters of the nonlinear mapping were set to $\alpha = 15$ and $\beta = 0.5$. It is remarkable that the three sets of results converge almost to the same distributions. This result is not a proof that the iteration process always converges to a single solution, and in fact the iteration has at least another trivial solution, for $u(x) = v(x)$ and $h(x) = \delta(0)$. However, this result shows to some extent the small dependency of this algorithm on the initial value.

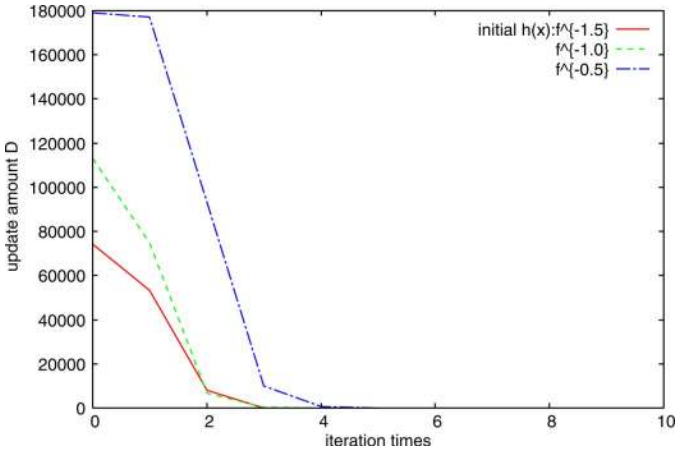


Fig. 9. Relationship between iteration times and update amount D.

As a measure of convergence of this algorithm, we define the update amount D:

$$D = \left\{ \sum_x \left(u^{(n)}(x) - u^{(n-1)}(x) \right)^2 \right\}^{1/2} \tag{21}$$

where $u^{(n)}(x)$ is the fundamental frequency distribution obtained at the nth iteration. The relationship between the iteration times and the update amount D for the cases of Fig. 6 is shown in Fig. 9. For all of three different initial $h(x)$, the update amount D decreases rapidly and at fifth iteration it becomes vanishingly small. This phenomenon is observed for almost all the other frames. The convergence of this algorithm is not guaranteed, but the convergence performance seems satisfying.

The nonlinear mapping function seems to attenuate not only the overtone components but also the fundamental frequency components with small amplitudes. The experiment result of two mixed sounds with significantly different amplitudes is shown in Fig. 10. The amplitude of the fundamental frequency component of G4 is quite smaller than that of C4, which is equal to $u_{\max}$, and therefore the nonlinear mapping function attenuates the smaller fundamental frequency component. However, after the deconvolution step the amplitude of the fundamental frequency component of G4 increase back to almost as large a value as it had in the original spectrum, and the nonlinear mapping function does not affect the small fundamental frequency component through the iteration as a whole. However, we learned from some experiments that the small fundamental frequency component is regarded as an harmonic component and suppressed when it is mixed with the large harmonic component of another fundamental frequency.

E. Multipitch Visualization

In addition to this framewise results, we can display the fundamental frequency distribution $u(x)$ as a time-frequency

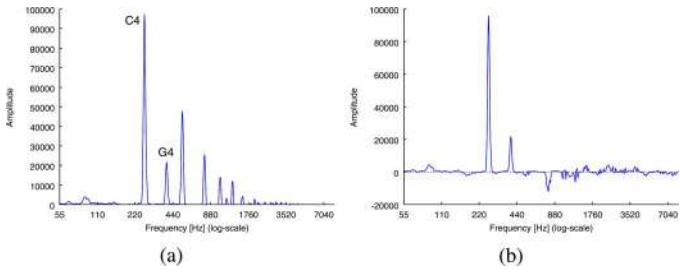


Fig. 10. Experimental result for two mixed sounds with significantly different amplitudes. (a) Wavelet transform of two mixed piano sounds (C4 and G4, excerpted from RWC Musical Instrument Sound Database [25]). (b) Result of specmurt analysis on (a).

plane. An example of pitch frequency visualization through specmurt analysis is shown in Fig. 11 (experimental conditions are the same to the evaluation in later section). We can see that the overlapping overtones in (a) are significantly suppressed by specmurt analysis in (b), which looks very close to the manually prepared piano-roll references in (c). Methods in which the pitch frequencies are parametrized can visualize the results as planes too, but the planes are reconstructed on the estimated frequency parameters, and the information about the number of sound sources is lost. In other words, these methods require the additional information to generate the planes, but the proposed method does not. Unlike these approaches, specmurt analysis generates a continuous fundamental frequency distribution and can enhance the spectrogram so that multiple fundamental frequencies become more visible without decision on the number of sound sources.

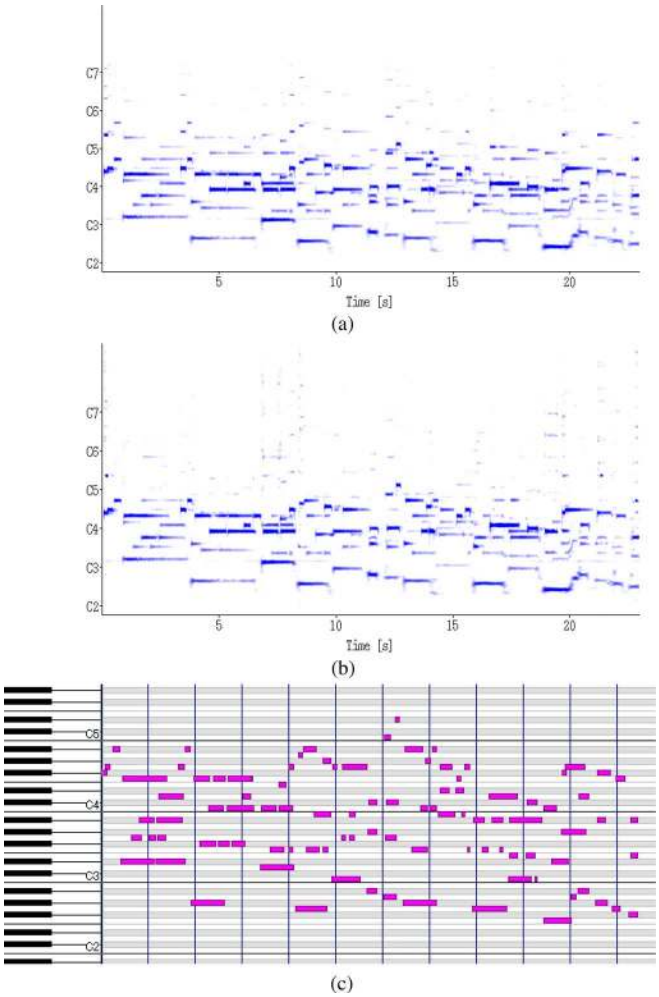


Fig. 11. Multipitch visualization of data 4, "For Two" (guitar solo) from the RWC Music Database, using specmurt analysis with quasi-optimized harmonic structure. (a) Log-frequency spectrum obtained through wavelet transform (input). (b) Estimated fundamental frequency distribution (output). (c) Piano-roll display of manually prepared MIDI data (reference). Overtones in (b) are fewer and thinner than in (a), and as a whole (b) is more similar to (c).

V. Experimental Evaluations

A. Conditions

Through iterative optimization of the common harmonic structure, improved performance is expected for automatic multipitch estimation. To experimentally evaluate the effectiveness of specmurt analysis for this purpose, we used 16-kHz sampled monaural audio signals excerpted from the RWC Music Database [27]. The estimation accuracy was evaluated by matching the analysis results with a reference MIDI data, which was manually prepared using the spectrogram as a basis, frame by frame. We chose this scheme because the duration accuracy of each note is also important. With note-wise matching, the duration cannot be evaluated and the evaluation result is affected more severely by instantaneous errors (for example one note can be divided into two notes by only one "OFF" error).

The RWC database also includes MIDI-format data, but they are unsuitable for matching: they contain timing inaccuracies from which the relevance of the computation of the accuracy from a frame-by-frame matching would strongly suffer. Furthermore, the durations of the MIDI reference are based on the musical notation in the score, and they do not reflect the real length of each sound signal, especially in the case of keyboard instruments, for which damping makes the offset harder to determine.

We chose HTC [28] and[1] PreFEst [11] for comparison. These methods are based on parametric models using the EM algorithm, in which power spectrum is fitted by weighted Gaussian mixture models. The common problem of the three methods is that the estimation result is not a "binary" data, i.e., the "active" or "silent" information, but some set of frequency, time, and amplitude. Moreover, the result of specmurt analysis has a continuous distribution with respect to frequency. In order to compare the reference MIDI data to the estimation results, we need to introduce some sort of thresholding process. This thresholding can have a large effect on the estimation accuracy, and the three methods produce three different types of output distribution. Therefore, we chose the highest accuracy among all the thresholds for each method.

We implemented a GUI editor to create a ground truth data set of pitch sequences as a MIDI reference (a screen-shot of

[1]Note that we implemented for the evaluation only the module called "PreFEst-core," a frame-wise pitch likelihood estimation, and not included the one called "PreFEst-back-end," a multiagent-based pitch tracking algorithm. Refer to [11] for their details.

Fig. 12. GUI for creating ground truth data of pitch sequences and calculating the best accuracy with three different algorithms (specmurt, HTC, and PreFEst) by changing the threshold value.

TABLE I
ANALYSIS CONDITIONS FOR THE LOG-FREQUENCY SPECTROGRAM

input signal	sampling rate	16kHz (monaural)
mother wavelet	type	Gabor function
	σ	50
	ω_0	1
frequency resolution γ (interval of a)		12.5 [cent/bin]
frame shift (interval of b)		16 [ms]

the GUI editor can be seen in Fig. 12). In this GUI, the music spectrogram is shown in the background and the user can generate a spectrogram-based reference with reliable duration. This system can also calculate the pitch estimation accuracy of the three methods for any threshold. The reference data made by this GUI are based on the bundled MIDI data and modified by hearing the audio and comparing to the spectrogram.

In our experiments, we set $N = 9$ and used a frequency characteristic of $f^{-1.5}$ as the initial common harmonic structure. As f^{-1} is generally understood as the most common frequency characteristic of natural sounds, $f^{-1.5}$ is a slightly "conservative" choice to avoid excess inverse filtering applied to the input wavelet spectrum. We empirically set $\alpha = 15$ and repeat the iterative steps five times throughout all data regardless of the fact that convergence is reached or not. Two values of the threshold magnitude parameter β, 0.2, and 0.5, were tested, as it seemed to have a significant effect on the estimation accuracy. Other analysis conditions for the log-frequency spectrogram are shown in Table I.

Table II shows the entire list of data, where approximately the first 20 s of each piece were used in our evaluation. Selection was made so as to cover some variety of timbre, solo/duet, instrument/voice, classic/jazz, but to exclude percussions.

The accuracy is calculated by frame-by-frame matching of the output and reference data. We define $O(n,t)$ as the (threshold-processed) output data, where n denotes the note number and t the time. $O(n_i,t_j)$ is 1 When the note number n_i is active at time t_j and 0 when it is not active. In the same way, $R(n,t)$ can be defined as the reference data, and the accuracy is calculated as follows:

$$\text{Accuracy}(\%) = \frac{\sum_j (X_j - D_j - I_j + S_j)}{\sum_j X_j} \times 100 \quad (22)$$

$$X_j : \sum_i R(n_i,t_j) \quad (23)$$

$$D_j : \sum_i R(n_i,t_j)(1 - O(n_i,t_j)) \quad (24)$$

$$I_j : \sum_i (1 - R(n_i,t_j))O(n_i,t_j) \quad (25)$$

$$S_j : \min\{D_j, I_j\}. \quad (26)$$

D_j denotes the number of deletion error, for which the output data is not active but the reference is active, and I_j denotes the number of insertion error, for which the output data is active but the reference is not active. However, both errors include the substitution errors, for which the output data is active at n_{i_1} but the reference is active at n_{i_2} (for example, a half-pitch error). Therefore, in order to avoid the double-count of substitution errors, we defined S_j as $\min\{D_j, I_j\}$ and the total error at t_j as $D_j + I_j - S_j$. This accuracy can be negative, and no compensation was given to unisono (i.e., several instruments play the same note simultaneously) and timbre. Of course, frame-by-frame matching produces a lower accuracy than note-by-note matching, and the result is hardly expected to reach 100% (e.g., even for a perfect note estimation accuracy, if all of the estimated note durations are half of the original, the calculated accuracy will be 50%).

B. Results

The experimental results are shown in Table III. First, when $\beta = 0.5$, for which overtone suppression is successfully done in Fig. 9, the accuracy results are averagely 2%–3% lower than for $\beta = 0.2$. One possible cause for that is the balance between the amplitudes of each note in a single frame. The nonlinear mapping with $\beta = 0.5$ has a larger attenuation effect, and therefore the estimation succeeds quickly in frames where the notes have about the same amplitude, otherwise notes with quite smaller amplitude are regarded as "noise" and suppressed.

For single-instrument data, the accuracy tends to be higher than for multiple-instrument data. Specmurt analysis assumes a common harmonic structure and this assumption is more justified for the spectrum of single-instrument music. Compared with previous works, the accuracy of the proposed method seems to be slightly lower than that of HTC, while it is almost equal to that of PreFEst.[2] However, the remarkable aspect of specmurt analysis is pitch visualization as a continuous distribution, and its advantage over the other algorithms is simplicity and quickness (it took 1.7 s with no iteration and 9.5 s with five iterations for 23.0-s length music data, including 1.2 s for wavelet transform). Hence, it is a very satisfying result that specmurt analysis earns a comparable score to previous state-of-the-art work.

[2]Note that multiple instrument data is also tested with a single prior distribution.

TABLE II
EXPERIMENTAL DATA FROM THE RWC MUSIC DATABASE [27]

	Data ID	Contents	Instrument(s)
data1	RWC-MDB-J-2001 No.1	Jazz – "Jive"	Piano
data2	RWC-MDB-J-2001 No. 2	Jazz – "For Two"	Piano
data3	RWC-MDB-J-2001 No. 6	Jazz – "Jive"	Guitar
data4	RWC-MDB-J-2001 No. 7	Jazz – "For Two"	Guitar
data5	RWC-MDB-J-2001 No. 8	Jazz – "Lounge Away"	Guitar
data6	RWC-MDB-J-2001 No. 9	Jazz – "Crescent Serenade"	Guitar
data7	RWC-MDB-C-2001 No.30	F. Chopin: Nocturne No. 2, in E♭ major, op. 9-2	Piano
data8	RWC-MDB-C-2001 No.35	E. Satie: Three Gymnopédies	Piano
data9	RWC-MDB-J-2001 No.12	Jazz – "For Two"	Flute + Piano
data10	RWC-MDB-C-2001 No.12	J. S. Bach: Ricercare à 6 from 'Musikalisches Opfer,' BWV 1079	Flute, Violin(1,2), Viola and Cello
data11	RWC-MDB-C-2001 No.39	C. Franck: Violin Sonata in A major. 4th mvmt.	Violin + Piano
data12	RWC-MDB-C-2001 No.42	C. Saint-Saëns: 'Le Cygne' from the Suite <Le Carnaval des Animaux>	Cello + Piano
data13	RWC-MDB-C-2001 No.44	N. Rimski-Korsakov: 'The Flight of the Bumble Bee'	Flute + Piano
data14	RWC-MDB-C-2001 No.49	G. Verdi: 'La donna è mobile qual piuma al vento' from <Rigoletto>	Tenor + Piano

TABLE III
ACCURACY RESULTS OF THE PROPOSED METHOD, HTC [28] AND PREFEST [11]

	proposed method		HTC[28]	PreFEst[11]
	$\beta = 0.2$	$\beta = 0.5$		
data1	59.0%	55.3%	**64.2%**	55.9%
data2	**63.9%**	62.8%	62.2%	62.3%
data3	51.3%	48.1%	**63.8%**	48.8%
data4	68.1%	64.4%	**77.9%**	71.8%
data5	67.0%	60.8%	75.2%	**76.2%**
data6	77.5%	76.8%	**81.2%**	74.2%
data7	57.0%	56.4%	**70.9%**	57.6%
data8	**63.6%**	64.4%	63.2%	53.6%
data9	44.9%	42.3%	43.2%	**48.2%**
data10	**48.9%**	45.9%	48.1%	47.7%
data11	**53.7%**	47.5%	50.6%	41.6%
data12	37.0%	34.2%	37.6%	**39.8%**
data13	**48.4%**	44.9%	43.2%	35.8%
data14	**35.8%**	35.7%	27.5%	33.0%
avg.	55.4%	52.8%	**57.8%**	53.3%
std. dev.	12.0%	12.2%	16.3%	13.9%

Some MIDI sounds are available at http://hil.t.u-tokyo.ac.jp/~lab/topics/SpecmurtSamples/.

VI. DISCUSSION

A. Comparison With Sparse Coding and Shifted NMF

Specmurt analysis utilizes the assumption that the harmonic structure is common among all the notes. In other words, specmurt analysis has a degree of freedom in the time direction but not in the frequency direction. In contrast, sparse coding in the frequency domain [14] expresses each note with one or more note-like representations, called *dictionary*. Assuming that any single sound spectrum can be represented using a single dictionary, sparse coding has a degree of freedom in the frequency direction but not in the time direction. Although a single note is in fact almost always expressed by multiple dictionaries, there is a similarity between specmurt analysis and sparse coding. Furthermore, the nonlinear mapping function in Section IV-A can be considered as a "sparseness" controller, in which the parameters α and β select the components which will "survive." In sparse coding, the objective function to optimize is expressed as a sum of a log-likelihood term (error between observation and model) and a log-prior term (sparseness constraints). In specmurt analysis, each step cannot be regarded as the optimization of the whole objective, but as the optimization of either term (Step 1 and Step 3 optimizing the likelihood term and Step 2 optimizing the sparseness term). It is no longer an "optimization," but at the expense of this, specmurt analysis accomplishes a simple and fast estimation.

Additionally, we will mention another method, shifted nonnegative matrix factorization [13]. In this method, a translation tensor is utilized, and any single sound is represented as a shifted-version of the frequency basis functions. Shifted nonnegative matrix factorization is very similar approach to specmurt analysis in terms of shift-invariant assumption, and this method can separate the sound sources performed by different musical instruments. However, the result is sensitive to the parameter of the number of allowable translations and the factorization does not utilize the harmonic structure constraint. As a result, the basis functions often include other than a single sound component or only a part of it, which can be also said of other NMF methods.

B. Practical Use of Specmurt Analysis

Specmurt analysis is based on a frame-by-frame estimation, and it is suitable for real-time applications. This method utilizes the assumption that the spectrum has a common harmonic structure, and therefore it cannot handle well nonharmonic sounds and the missing fundamental.

One problem concerning the iterative estimation in specmurt analysis is the stability of the harmonic structure as an inverse filter. Even if the harmonic structure is properly estimated, there is a possibility that the Fourier transform of the harmonic structure $H(y)$ has zero (or near zero) values. An example is shown in Fig. 13. The wavelet spectrum Fig. 13(a) is excerpted from the spectrogram of data 2 in Table II. The estimated common harmonic structure is Fig. 13(b) and seems to be estimated properly, but the estimated fundamental frequency distribution fluctuates heavily. This is because $H(y)$ has a near zero value at a certain point in the y domain, and the inverse filter response $h^{-1}(x)$ (shown in Fig. 13(d)) has a large sinusoid component. The relationship between the harmonic structure coefficients and the stability of the inverse filter is not completely clear yet, but it seems to occur when a new sound starts. These errors occur at very few frames so that they do not affect so much the estimation result as a whole, and they could be detected through the heuristic approach, such as watching the absolute value of the inverse filter, for example. However, as a future work we will

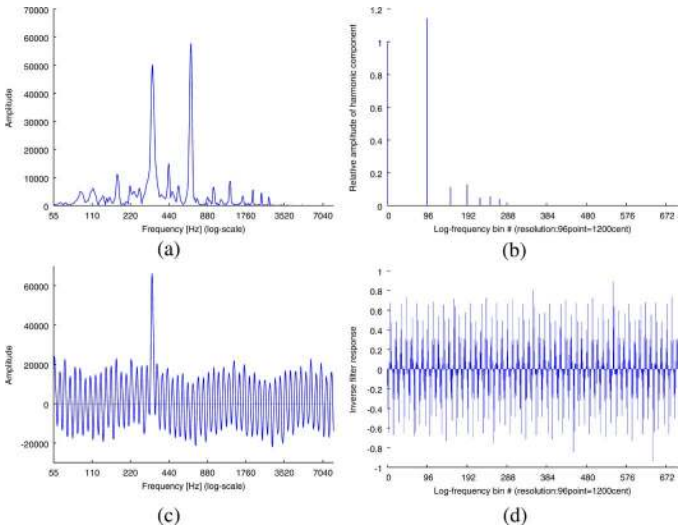


Fig. 13. Example of division by zero in (11) and its influence on $u(x)$. (a) Wavelet spectrum $v(x)$. (b) Estimated common harmonic structure pattern $h(x)$. (c) Estimated fundamental frequency distribution $u(x)$. (d) Inverse filter response $h^{-1}(x)$.

need to investigate the behavior of the inverse filter generated from the common harmonic structure.

VII. Conclusion

We presented a novel nonlinear signal processing technique called specmurt analysis which is parallel to cepstrum analysis. In this method, multiple fundamental frequencies of a polyphonic music signal are detected by inverse filtering in the log-frequency domain and represented in a piano-roll-like display. Iterative optimization of the common harmonic structure was also introduced and used in sound-to-MIDI conversion of polyphonic music signals.

Future work includes the extension of specmurt analysis to a 2-D approach, the use of specmurt analysis to provide initial values for precise multipitch analysis based on harmonically constrained Gaussian mixture models [28], application to automatic transcription of music (sound-to-score conversion) through combination with rhythm transcription techniques [29], music performance analysis tools, and interactive music editing/manipulation tools.

Appendix

To prove the nonsingularity of the matrix in (18), which we denote by $\mathbf{A}$, we need to show that there is no nonzero vector $\mathbf{e}$ satisfying

$$\mathbf{Ae} = \mathbf{0} \tag{27}$$

or

$$\sum_{k=2}^{N} e_k \int_{-\infty}^{\infty} \bar{u}(x-\Omega_j)\bar{u}(x-\Omega_k)dx = 0 \quad (\forall j, 2 \leqq j \leqq N) \tag{28}$$

where $\mathbf{e} = (e_2, \ldots, e_N)$. If one could find such a vector, it would of course also satisfy $\mathbf{e}^T\mathbf{Ae} = 0$. Then, from (18) and the special form of the coefficients $a_{j,k}$ in (19), we get

$$\mathbf{e}^T\mathbf{Ae} = \sum_{i,j} \int_{-\infty}^{\infty} e_i e_j \bar{u}(x-\Omega_i)\bar{u}(x-\Omega_j)dx \tag{29}$$

$$= \int_{-\infty}^{\infty} \left\{ \sum_i e_i \bar{u}(x-\Omega_i) \right\}^2 dx. \tag{30}$$

Thus, if $\mathbf{e}^T\mathbf{Ae} = 0$, then

$$\forall x \in (-\infty, +\infty),\ z(x) = \sum_i e_i \bar{u}(x-\Omega_i) = 0. \tag{31}$$

We assume that $\bar{u}(x)$ has a limited support I (which is obviously justified for a fundamental frequency distribution) and that $x_{inf} = \inf\{x|\bar{u}(x) \neq 0\}$ and $x_{sup} = \sup\{x|\bar{u}(x) \neq 0\}$ can thus be defined. Then, the supports of the shifted versions $\bar{u}(x-\Omega_i)$ are $I+\Omega_i$. Moreover, for all $2 \leq i \leq N-1$, we have

$$\Omega_{i+1} - \Omega_i = C \log\left\{\frac{(i+1)}{i}\right\} \tag{32}$$

$$\geq C \log\left\{\frac{N}{(N-1)}\right\} = \eta\,. \tag{33}$$

By definition of x_{inf}, there exists $x_0 \in [x_{inf}, x_{inf}+\eta)$ such that $\bar{u}(x_0) \neq 0$.

If we consider $z(x_0+\Omega_2)$, we see that $\bar{u}(x_0+\Omega_2-\Omega_i)$ is nonzero for $i = 2$ and zero for $i > 2$. Thus, $e_2 = 0$. By then considering consecutively $z(x_0+\Omega_3), \ldots, z(x_0+\Omega_N)$, we show similarly that $e_3 = 0, \ldots, e_N = 0$. Therefore, (27) holds if and only if $\mathbf{e}$ is a zero vector and the proof is complete. In computational calculation, the same can be said as long as the frequency resolution is high enough for x_0 to exist.

Acknowledgment

The authors would like to thank Dr. N. Ono and Mr. J. Le Roux for valuable discussion about the Appendix .

References

[1] B. P. Bogert, M. J. R. Healry, and J. W. Tukey, "The quefrency alanysis of time series for echos: Cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe-cracking," in *Proc. Symp. Time Series Analysis*, 1963, pp. 209–243.

[2] A. M. Noll, "Short-time spectrum and 'cepstrum' techniques for vocal-pitch detection," *J. Acoust. Soc. Amer.*, vol. 36, no. 2, pp. 296–302, Feb. 1964.

[3] S. Sagayama and F. Itakura, "On individuality in a dynamic measure of speech," in *Proc. ASJ Conf.* (in Japanese), Jul. 1979, pp. 589–590.

[4] S. E. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[5] S. Imai and T. Kitamura, "Speech analysis synthesis system using log magnitude approximation filter," (in Japanese) *Trans. IEICE Japan*, vol. J61-A, no. 6, pp. 527–534, 1978.

[6] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 806–816, Nov. 2003.

[7] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka, "Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism," in *Proc. Int. Joint Conf. Artif. Intell.*, 1995, vol. 1, pp. 158–164.

[8] J. C. Brown, "Musical fundamental frequency tracking using a pattern recognition method," *J. Acoust. Soc. Amer.*, vol. 92–3, pp. 1394–1402, 1992.
[9] M. Goto, "A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Jun. 2000, vol. 2, pp. 757–760.
[10] M. Goto, "A predominant-F0 estimation method for CD recordings: MAP estimation using EM algorithm for adaptive tone models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Sep. 2001, vol. 5, pp. 3365–3368.
[11] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Commun.*, vol. 43, no. 4, pp. 311–329, 2004.
[12] F. Sha and F. Saul, "Real-time pitch determination of one or more voices by nonnegative matrix factorisation," in *Proc. Neural Inf. Process. Syst.*, 2004, pp. 1233–1240.
[13] D. FitzGerald, M. Cranitch, and E. Coyle, "Shifted non-negative matrix factorization for sound source separation," in *IEEE Workshop Statist. Signal Process.*, 2005, pp. 1132–1137.
[14] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 179–196, Jan. 2006.
[15] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 50–57, Jan. 2006.
[16] S. Godsill and M. Davy, "Bayesian harmonic models for musical pitch estimation and analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2002, vol. 2, pp. 1769–1772.
[17] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2002, vol. 2, pp. 1757–1760.
[18] A. Klapuri, T. Virtanen, and J. Holm, "Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals," in *Proc. COST-G6 Conf. Digital Audio Effects*, 2000, pp. 233–236.
[19] H. Kameoka, T. Nishimoto, and S. Sagayama, "Extraction of multiple fundamental frequencies from polyphonic music," *Proc. Int. Congr. Acoust.*, pp. 59–62, 2004.
[20] H. Kameoka, T. Nishimoto, and S. Sagayama, "Separation of harmonic structures based on tied Gaussian mixture model and information criterion for concurrent sounds," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 4, pp. 297–300.
[21] T. Irino and R. D. Patterson, "Segregating information about the size and shape of the vocal tract using a time-domain auditory model: The stabilised wavelet–Mellin transform," *Speech Commun.*, vol. 36, no. 3, pp. 181–203, 2002.
[22] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis – A unified approach to speech spectral estimation," in *Proc. Int. Conf. Spoken Lang. Process.*, 1994, pp. 1043–1046.
[23] S. Saito and F. Itakura, "The theoretical consideration of statistically optimum methods for speech spectral density," (in Japanese) Elec. Commun. Lab., NTT, Tokyo, Japan, 1966, Tech. Rep. 3107.
[24] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals," in *Proc. Int. Conf. Speech Commun. and Process.*, 1967, pp. 360–361.
[25] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. Int. Conf. Music Inf. Retrieval*, Oct. 2003, pp. 229–230.
[26] S. Saito, H. Kameoka, N. Ono, and S. Sagayama, "Iterative multipitch estimation algorithm for MAP specmurt analysis," (in Japanese) IPSJ SIG Tech. Rep., Aug. 2006, vol. 2006-MUS-66, pp. 85–92.
[27] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music database," in *Proc. Int. Symp. Music Inf. Retrieval*, Oct. 2002, pp. 287–288.
[28] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 982–994, Mar. 2007.
[29] H. Takeda, T. Nishimoto, and S. Sagayama, "Automatic rhythm transcription from multiphonic MIDI signals," in *Proc. Int. Conf. Music Inf. Retrieval*, Oct. 2003, pp. 263–264.

Shoichiro Saito (S'06) received the B.E. and M.E. degrees from the University of Tokyo, Tokyo, Japan, in 2005 and 2007, respectively.

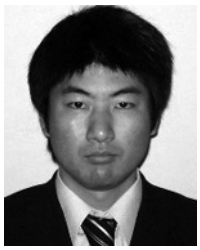
He is currently a Research Scientist at NTT Cyber Space Laboratories, Tokyo, Japan. His research interests include music signal processing, speech analysis, and acoustic signal processing.

Mr. Saito is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), Japan, Information Processing Society of Japan (IPSJ), and Acoustical Society of Japan (ASJ).

Hirokazu Kameoka (S'05) received the B.E., M.E., and Ph.D. degrees from the University of Tokyo in Tokyo, Japan, in 2002, 2004, and 2007, respectively.

He is currently a Research Scientist at NTT Communication Science Laboratories, Atsugi, Japan. His research interests include computational auditory scene analysis, acoustic signal processing, speech analysis, and music application.

Dr. Kameoka is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), Information Processing Society of Japan (IPSJ), and Acoustical Society of Japan (ASJ). He was awarded the Yamashita Memorial Research Award from IPSJ, Best Student Paper Award Finalist at the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), the 20th Telecom System Technology Student Award from the Telecomunications Advancement Foundation (TAF) in 2005, the Itakura Prize Innovative Young Researcher Award from ASJ, 2007 Dean's Award for Outstanding Student in the Graduate School of Information Science and Technology from the University of Tokyo, and the 1st IEEE Signal Processing Society Japan Chapter Student Paper Award in 2007.



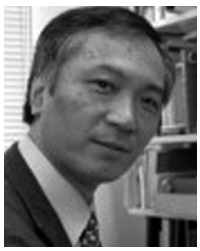
Keigo Takahashi received the B.E. and M.E. degrees from the University of Tokyo, Tokyo, Japan, in 2002 and 2004, respectively.

He is currently a Technical Official at the Community Safety Bureau, National Police Agency. His research interests include musical signal processing, music application, and speech recognition.

Takuya Nishimoto received the B.E. and M.E. degrees from Waseda University, Tokyo, Japan, in 1993 and 1995, respectively.

He is a Research Associate at the Graduate School of Information Science and Technology, University of Tokyo. His research interests include spoken dialogue systems and human–machine interfaces.

Mr. Nishimoto is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), Japan, Information Processing Society of Japan (IPSJ), Acoustical Society of Japan (ASJ), Japanese Society for Artificial Intelligence (JSAI), and Human Interface Society (HIS).



Shigeki Sagayama (M'82) was born in Hyogo, Japan, in 1948. He received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1972, 1974, and 1998, respectively, all in mathematical engineering and information physics.

He joined Nippon Telegraph and Telephone Public Corporation (currently, NTT) in 1974 and started his career in speech analysis, synthesis, and recognition at NTT Laboratories, Musashino, Japan. From 1990 to 1993, he was Head of the Speech Processing Department, ATR Interpreting Telephony Laboratories, Kyoto, Japan, pursuing an automatic speech translation project. From 1993 to 1998, he was responsible for speech recognition, synthesis, and dialog systems at NTT Human Interface Laboratories, Yokosuka, Japan. In 1998, he became a Professor of the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan. In 2000, he was appointed Professor of the Graduate School of Information Science and Technology (formerly Graduate School of Engineering), University of Tokyo. His major research interests include processing and recognition of speech, music, acoustic signals, hand writing, and images. He was the leader of anthropomorphic spoken dialog agent project (Galatea Project) from 2000 to 2003.

Prof. Sagayama is a member of the Acoustical Society of Japan (ASJ), Institute of Electronics, Information, and Communications Engineers (IEICEJ) Japan, and Information Processing Society of Japan (IPSJ). He received the National Invention Award from the Institute of Invention of Japan in 1991, the Chief Official's Award for Research Achievement from the Science and Technology Agency of Japan in 1996, and other academic awards including Paper Awards from the IEICEJ in 1996 and from the IPSJ in 1995.