ELSEVIER

Speech Communication 27 (1999) 337–349

SPEECH COMMUNICATION

A sound source identification system for ensemble music based on template adaptation and music stream extraction [1]

Kunio Kashino *, Hiroshi Murase

NTT Basic Research Laboratories, 3-1 Morinosato-Wakamiya, Atsugi-shi, Kanagawa 243-0198, Japan

Received 30 December 1997; received in revised form 27 September 1998

Abstract

Sound source identification is an important problem in auditory scene analysis when multiple sound objects are simultaneously present in the scene. This paper proposes an adaptive method for sound source identification that is applicable to real performances of ensemble music. For musical sound source identification, the feature-based methods and template-matching-based methods were already proposed. However, it is difficult to extract features of a single note from a sound mixture. In addition, sound variability has been a problem when dealing with real music performances. Thus this paper proposes an adaptive method for template matching that can cope with variability in musical sounds. The method is based on the matched filtering and does not require a feature extraction process. Moreover, this paper discusses musical context integration based on the Bayesian probabilistic networks. Evaluations using recordings of real ensemble performances have revealed that the proposed method improve the source identification accuracy from 60.8% to 88.5% on average. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Sound source identification; Music recognition; Template adaptation; Music stream; Probablistic network

1. Introduction

In recent years scene analysis based on acoustic information, termed auditory scene analysis, has received renewed interest. Auditory scene analysis means recognizing many acoustic events occurring simultaneously (Bregman, 1990; Cooke, 1991). From the engineering point of view, a specific feature of the auditory scene analysis problem is that noise and signals are not defined uniquely in advance; a computer system for auditory scene analysis must handle multiple 'signals' simultaneously. This is in contrast to problem definitions in the conventional sound recognition tasks, such as speech recognition, where the signal is defined as human speech and noise is defined as all non-speech sounds.

In order to deal with many simultaneous signals, sound source separation problems have been addressed since as early as the 1970s. Approaches using microphone arrays have been one of the major research streams (Mitchell et al., 1971; Flanagan et al., 1985), and harmonic selection is another major method (Parsons, 1976; Nehorai and Porat, 1986). A hybrid approach that integrates the microphone-array approach and the harmonic selection approach is also found in the literature (Nakatani et al., 1995). In addition,

* Corresponding author. Tel.: +81-462-40-3568; fax: +81-462-40-4708; e-mail: kunio@ca-sun1.brl.ntt.co.jp

[1] Speech files available. See http://www.elsevier.nl/locate/specom.

PII: S0167-6393(98)00078-8

independent component analysis has applied to sound source separation by many authors under the term of blind source separation (Cardoso, 1989; Bell and Sejnowski, 1995; Lee et al., 1997).

However, sound source separation is only half of the auditory scene analysis problem. The other half is sound source identification, which means recognizing the name or label of each acoustic event. In comparison with sound source separation, relatively limited amount of work has been reported on this problem (Lesser et al., 1993; Ellis, 1996).

Thus this paper addresses the sound source identification problem for sound mixtures. Ensemble music was chosen as an example target domain. The problem is to recognize the name of a musical instrument playing each musical note. This identification will be necessary in application systems such as automatic music transcription systems and signal-to-MIDI (Musical Instrument Digital Interface) conversion system. We consider that the present work for ensemble music is a step towards auditory scene analysis in general.

The approach towards the music recognition task has also had a long history. Early work inspired by frequency-analysis techniques concerns the transcription of a single-pitched melody such as a vocal solo (Piszczalski and Galler, 1977; Niihara and Inokuchi, 1986). Later, recognition systems for multiple-pitched music performed by a single musical instrument (e.g., piano solos) were proposed (Katayose and Inokuchi, 1989). However, few works have addressed the recognition of multiple-pitched music performed by *multiple* kinds of musical instruments (e.g., such as by a chamber ensemble), although several attempts can be found in the literature (Mont-Reynaud, 1985; Chafe and Jaffe, 1986; Brown and Cooke, 1994; Kashino et al., 1995a). Consequently, music recognition systems which can deal with ensemble music played by many musical instruments simultaneously, and give reasonable accuracy, have not yet been realized.

For the sound source identification problem, the approach first considered may be one based on the timbre feature such as discriminant analysis. For example, Cosi et al. (1994) proposed a method to recognize musical timbre based on an auditory model and the self-organizing neural network model. Brown and Cooke (1994) reported a timbre classification method using a two-dimensional timbre space. When applied to real music performances, however, methods that use only timbre features may not sufficiently accurate. This is because multiple notes are simultaneously played most of the time in music, and therefore frequency components from different instruments overlap. When the components overlap, it is difficult to extract the timbre feature for each note.

Thus a template-matching-based approach, which is based on matching between an input spectrum pattern and a mixture of the spectrum pattern of each note stored in advance, and a hybrid method of the template-matching-based approach and the discriminant analysis have been proposed (Kashino and Tanaka, 1993; Kashino et al., 1995a,b). Although these matching-based methods alleviate the harmful impacts caused by the overlapping frequency components, the methods are adversely affected by variations of timbres caused by individual differences of musical instruments and expressions of performances. In fact, the music recognition system proposed by Kashino et al. (1995a) has been applied only to artificial performances synthesized by a sampler. [2]

Thus this paper proposes the template adaptation technique for an adaptive template matching. This adaptation is to cope with variability of a real musical signal due to the individual differences of musical instruments and musical expressions. This paper also discusses integration of musical context. Adaptive template matching uses only local information, and therefore matching results can still be ambiguous. The basic idea of musical context integration is to extract the streams of melodies and utilize them in order to improve the accuracy of sound source identification.

The rest of this paper is organized as follows. Section 2 overviews the processing architecture. Section 3 discusses template adaptation. Section 4 then proposes a method of constructing a graph,

[2] A sampler is an electronic musical instrument that stores the waveforms of various musical instruments and plays them back on MIDI input from a computer.

which we call MSN (music stream network), that represents the streams of melodies. Section 5 further discusses how to update a posteriori probability for each sound source using the MSNs. Section 6 presents some examples of system behaviors and evaluates the accuracy of the processing. Finally Section 7 will conclude this paper.

2. System architecture

2.1. Overview

Since a sound source identification system must deal with many 'signals' simultaneously, it is natural to build a system by accumulating processing modules, each of which tries to identify a specific target sound as a 'signal', regarding other sounds as 'noise'. Such modules should interact with each other to create a valid interpretation of signals. This is because the multiple interpretations made by the multiple modules may conflict with each other.

Thus we propose a system architecture based on a multi-agent scheme, as shown in Fig. 1. An agent is a simple-functioning processing module that interacts with each other (Maes, 1990). Here the agent function is detecting and identifying a specific sound in charge from a mixture.

The sound source identification system assumes that the input is an ensemble music signal divided into frames and a list of fundamental frequencies included in each frame. As output, the system creates a symbolic representation that is similar to

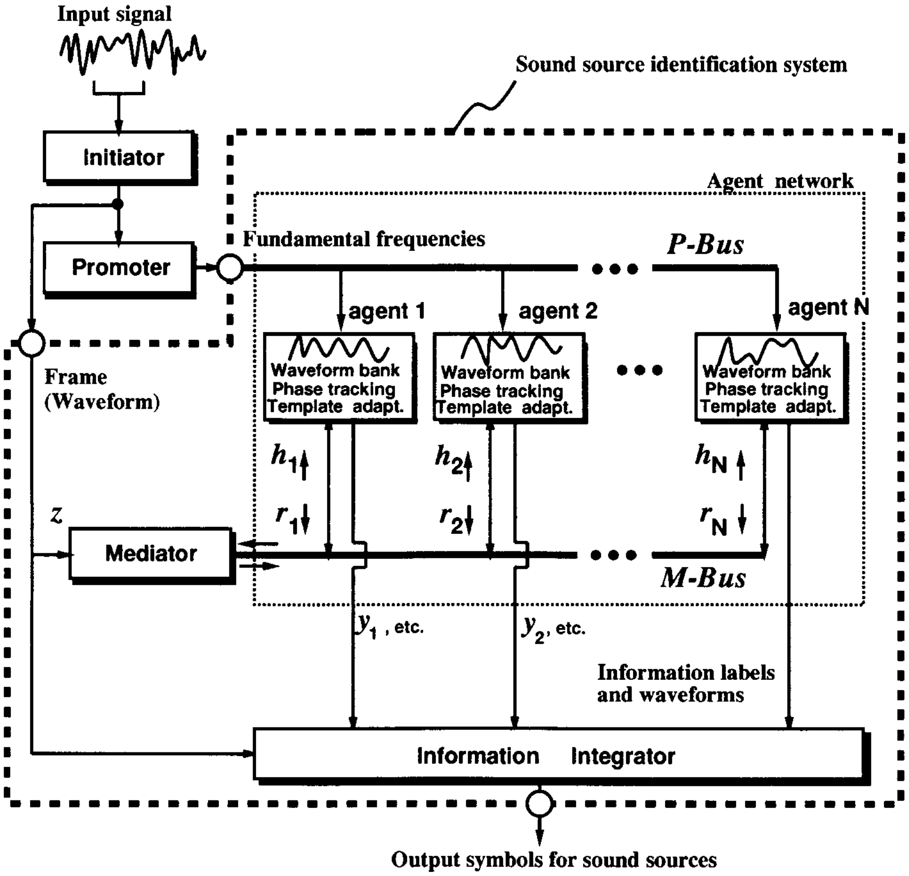

Fig. 1. The Ipanema architecture.

a musical score. The input frame length is not fixed; it is assumed that a new frame is created with each new onset of a musical note.

The sound source identification system consists of the agent network, a mediator of the agents, and an information integrator. However, our current implementation includes an onset detection module (called an initiator) and a fundamental frequency extraction module (called a promoter), as shown in Fig. 1. The initiator divides an input signal into frames and sends the waveform of each frame for subsequent processing. It tries to find the onset of a sound and creates the frames. The promoter performs frequency analysis on the waveform received from the initiator and extracts fundamental frequency components. There may be many of these. This module is called a promoter because it assists the agent activities. We call the architecture shown in Fig. 1, including the initiator and the promoter, Ipanema.

2.2. Agents

In our architecture, each agent in the agent network is a processing module that corresponds to a single sound source (a flute, for example). Each agent maintains a bank of waveforms, each of which is a waveform of a single note of a specific pitch and expression.

Each agent examines the input fundamental frequencies and checks whether the frequency is within the pitch range of the sound source corresponding to the agent. If the agent i judges that there is a possibility of being included, then the agent suggests a waveform r_i, applying a phase tracking method to one of the waveforms stored in the bank, as discussed in Section 3. If the agent infers little possibility of being included, then it takes no action.

The agents that suggested waveforms then modify them to minimize the squared error between the input signal and the sum of the suggested waveforms. To do this, the r_i waveforms are written to the common M-Bus (mediation bus) and passed to the mediator. The agents then wait for the mediator to send back answers. The answers of the mediator are sets of filter coefficients that optimally modify r_i. Each agent reads the answer from the mediator via the M-Bus and then generates an FIR filter with the returned coefficients to calculate a modified waveform y_i. These mechanisms will be further discussed in Section 2.3.

The final output of the agent is the waveform y_i and a predetermined information label for the waveform; for example, 'Flute C4'.

2.3. Mediator

As explained in Section 3, here mediation of agents is reduced to the problem of matrix calculation. Thus the mediator first receives the r_i from agents via the M-Bus. It then calculates the optimal filter coefficients for each agent (Eq. (4)). Finally the mediator sends the coefficients back to each agent using the M-Bus.

2.4. Information integrator

The information integrator is a post processing module that revises the output of the system. It receives an information label and an output waveform from each agent and judges which sound sources are present. Basically, the judgment is made based on the correlation between the agent output waveform and an input signal. However, since the initiator, promoter, and the agents in the agent network operate on a frame-by-frame basis and only local information is used, the matching results can still be ambiguous. Therefore, the information integrator treats this correlation as hypotheses rather than final results, and verifies these hypotheses by integrating musical context. The mechanism of this information integration is discussed in Sections 4 and 5.

3. Template adaptation

This section focuses on the template adaptation calculated by the agent network and the mediator.

3.1. Template filtering

An input acoustic signal $z(k)$ is represented as a sum of waveforms $y_n(k)$, where n specifies sound source and k represents time. Our problem can be formulated as minimization of J in the equation

$$J = E\left[\left\{z(k) - \sum_{n=0}^{N-1} y_n(k)\right\}^2\right], \tag{1}$$

where N is the estimated number of sound sources, which is not predefined, and E denotes the temporal average. For $y_n(k)$, we employ one of the simplest models as depicted in Fig. 2. The model can be written as

$$y_n(k) = \sum_{m=0}^{M-1} h_n(m)\, r_n(k-m), \tag{2}$$

where h is the filter impulse response, r is a template waveform, and M is the length of impulse response length, that is, the number of taps when H is an FIR filter.

In this model, the fixed sets of h and r cannot be predetermined. This is because there is a diversity of waveforms even for one specific sound source. Consider the example of musical instruments in Fig. 3. Both waveforms (a) and (b) are piano sounds. Both waveforms are different in terms of both the phase and spectral power information. Therefore we need an adaptive mechanism. Eq. (1) is rewritten using Eq. (2) as

$$J = E\left[\left\{z(k) - \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} h_n(m)\, r_n(k-m)\right\}^2\right]. \tag{3}$$

The necessary condition for J to hold the minimum value over $h_n(m)$ is that the partial derivatives $\partial J/\partial h_n(m)$ are 0 for all n and m. Using this condition, it is straightforward to derive $N \times M$ simultaneous linear equations as follows:

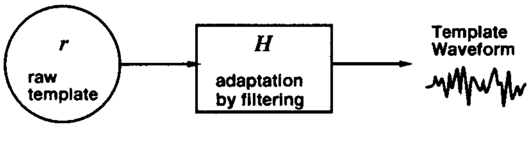

Fig. 2. A sound source model that consists of a template r and an FIR filter H. Here, H modifies the waveform of the original template r to cope with variation of a sound.

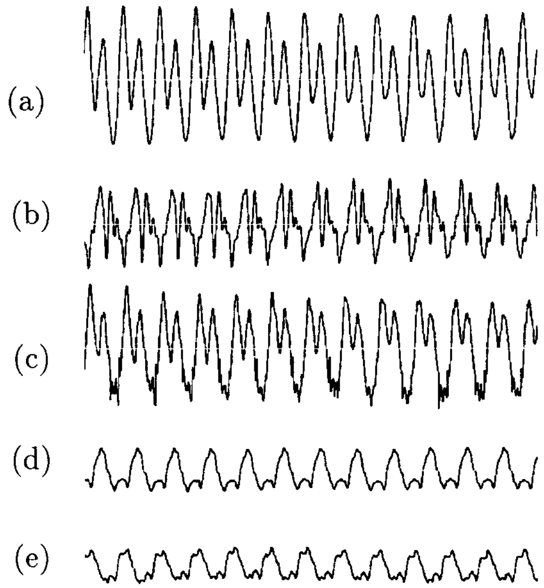

Fig. 3. Template filtering. (a) The input signal from a Yamaha piano. It also shows the F4 note between 160 and 195 ms after onset; (b) the original piano template (which is Boesendolfer's) with the same pitch and time portion as (a); (c), (d) and (e) templates processed by the template filtering method. They are produced by a piano, flute and violin template, respectively; (c) is obtained by filtering waveform (b). Note that (c) has a higher correlation with (a) than either (d) or (e). In all cases, a 160 tap filter and 48 kHz sampling rate was used.

$$\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} E[r_i(k-j)\, r_n(k-m)]\, h_n(m) = E[r_i(k-m)\, z(k)]\,, \tag{4}$$

where $i = \{0, 1, \ldots, N-1\}$ and $j = \{0, 1, \ldots, M-1\}$. Since the number of equations $(N \times M)$ equals the number of unknown parameters $(h_n(m))$, the problem is reduced to the inverse matrix calculation.

3.2. Phase tracking

For the above optimization scheme to be effective, the fundamental frequency of each template r must be exactly the same as the frequency included in z. This is because a linear filter, H, cannot change the frequency of an input signal. Therefore we need a phase tracking (i.e. instantaneous frequency tracking) method, which changes the phase of template r in accordance with the phase of the corresponding sound source signal included in the input signal z.

If the input signal is not a mixture of multiple sounds but a single sound, adaptive pitch tracking methods already invented can be used. However, such signal processing methods are not directly applicable to a sound mixture where multiple pitches are present. Thus we have devised a simple algorithm to implement phase adaptation. The algorithm consists of the following five steps:

1. For each fundamental frequency component given in the input, choose q_i which is a possible template for a sound included in z. As mentioned earlier, this choice is based on the pitch range of the instrument.
2. Apply a narrow-band bandpass filter to q_i, using the average fundamental frequency of each q_i as the bandpass filter center frequency. Here it is assumed that the fundamental frequency fluctuation is small with regard to the bandpass filter Q value. For each time sample, store the phase of the output waveform of the bandpass filter. Let $p_{q,i}(k)$ denote the phase at time k.
3. Apply the same bandpass filter, as applied to q_i, to the input z, and store the phase information for each fundamental frequency as $p_{z,i}(k)$.
4. Calculate the required time shift $\Delta k_{r,i}(k)$. Because the phase difference $\Delta p_{q,i}(k)$ is given by

   $$\Delta p_{r,i}(k) = p_{z,i}(k) - p_{q,i}(k), \tag{5}$$

   the time shift $\Delta k_{q,i}(k)$ is calculated using

   $$\Delta k_{q,i}(k) = \frac{f_s}{2\pi f_{c,i}} \Delta p_{q,i}(k). \tag{6}$$

   f_s is the sampling frequency and $f_{c,i}$ is the applied bandpass filter center frequency.
5. The modified amplitude r_i at time k is given by

   $$r_i(k) = q_i(\, k - \Delta k_{q,i}(k)). \tag{7}$$

This algorithm is depicted in outline in Fig. 4.

As discussed so far, the proposed method decomposes the sound variability into two factors: (1) the fundamental frequency fluctuations and (2) overtone phase and amplitude fluctuations. Firstly, the phase tracking method absorbs the former fluctuations and then the template filtering deals with the latter fluctuations.

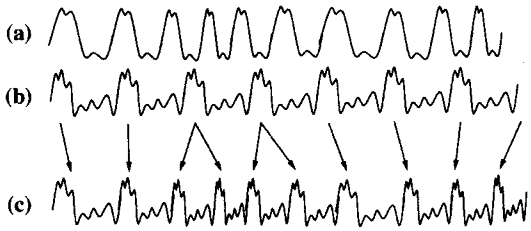

Fig. 4. Waveforms demonstrating phase tracking. (a) The input waveform z; (b) a template before phase tracking q_i; (c) a template after phase tracking r_i.

4. Music stream extraction

When a listner is presented with an ensemble music performance by a flute and piano, it will usually not be difficult for the listener to answer that the performance is performed by a flute and piano, even if the listner has not experienced special music training. However, if the presented music is a short-time fragment (0.3 s for example), the task becomes difficult. This implies that a human listener uses musical context to interpret the acoustic signals.

There are several ways to integrate musical context information for sound source identification; for example, a probabilistic network approach. The probabilistic method is widely used in the speech recognition field and proven to be very powerful in representing temporal transitions of phonemes or words.

Here we consider a Bayesian probabilistic network to represent a melody stream. The Bayesian network formulated by Pearl (1986) is a tool for calculating the a posteriori probabilities when a series of events related to each other are observed. It has already applied successfully to music recognition (Kashino et al., 1995b).

Consider two musical notes n_k, n_{k-1} (k denotes the order of the onset times of these notes, n_{k-1} precedes n_k). We define $Z(n_k, n_{k-1})$ using

$$Z(n_k, n_{k-1}) = W \sum_i \{ - w_i \log \; P_i(n_k, n_{k-1}) \}, \tag{8}$$

where i is a suffix that enumerates the factor of Z, P_i is a conditional probability of the occurrence of the n_{k-1} to n_k transition in a given musical context, and w_i (> 0) is a weight for each factor. Since the component $-\log P_i$ is self-information delivered by the transition from n_{k-1} to n_k, Z can be viewed as a weighted sum of self-information. Thus Z reflects the infrequency of the transition for these two notes. Therefore we define the 'music stream' as the sequence of musical notes that gives a local minimum of Z.

The term W is a time window that is defined as

$$W(\delta t) = \exp\left(\frac{\delta t}{\tau}\right), \tag{9}$$

where δt is the difference of onset times of these two notes, and τ is a time constant. Unlike ordinary time windows, W becomes greater as δt increases.

In this paper, the following three factors of Z are considered: (P_1) transition probabilities of musical intervals, (P_2) transition probabilities of timbres, and (P_3) transition probabilities of musical roles. These factors are now discussed.

4.1. Musical interval transitions

In tonal music, musical intervals of note transitions do not appear equally; some intervals are more frequent than others. Thus the pitch transition probability in a melody can be utilized as P_1 in Eq. (8). To obtain P_1, we analyzed 397 melodies extracted from 196 pop scores and 201 jazz scores, and calculated the probabilities of musical intervals. The number of note transitions was 62 689. Fig. 5 shows the obtained probabilities. The analysis was made only for principal melodies and may not be precisely valid for the other melodies such as bass-lines or the parts arranged for polyphonic instruments such as pianos. For simplicity, however, we used probabilities shown in Fig. 5 as P_1 for all cases.

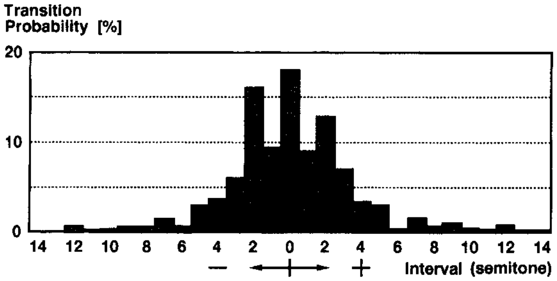

Fig. 5. Probabilities of musical intervals.

4.2. Timbre similarity

It is reasonable to suppose that a sequence of notes tends to be composed of notes that have similar timbres. To incorporate this tendency, we define a distance measure between the timbres of two notes and estimate the probabilities that two notes that are a certain distance apart sequentially appear in a music stream. These probabilities form P_2 in Eq. (8).

The distance between timbres is defined as the Euclidean distance between the timbre vectors. A timbre vector is a vector whose elements are the activity levels of agents, that is, the correlation values between the output from the agents and the corresponding portion of the input signal. Thus the number of elements of the timbre vector equals the number of agents in the agent network including non-activated agents. The distances between successive notes in a sequence are translated into probabilities using a normalized histogram as explained in Fig. 6. This histogram models the distribution of timbre vectors for notes. In the current implementation, we assumed the normal distribution with an empirically determined standard deviation instead of statistically creating the histogram based on the timbre vector sampling.

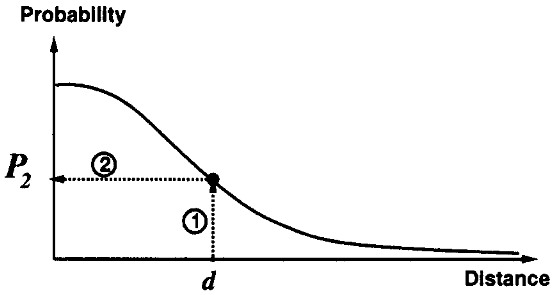

Fig. 6. Conversion from distance to probability. The calculated distance d is converted to P_2, which is the probability of appearance of the distance in a sequence of musical notes, by using a histogram of distances. The histogram is normalized so that the total degrees sum to one.

4.3. Musical role consistency

In ensemble music, a sequence of notes can be regarded as carrying a musical role such as a principal melody or a bass-line. To introduce such musical semantics, we evaluate the probability P_3 that a note plays a musical role in a sequence of notes. Specifically, here we consider the highest pitch and the lowest pitch in simultaneous notes as 'roles'. For simplicity, we empirically approximated P_3:

$$P_3 = ar + b, \tag{10}$$

where a and b are constants, and r is the rate of the highest (or lowest) notes in the music stream under consideration. Eq. (10) represents a musical heuristic that the music stream formed by the highest (lowest) notes tends to continue to flow to the highest (lowest) note.

4.4. Creating MSNs

Using Eq. (8), the networks that correspond to sequences of musical notes, which we call music stream networks (MSNs), are built by the following procedure. Fig. 7 shows the status of nodes when the new node n_k was just created. Firstly, possible links terminated at a new node is considered; each time a new node is created, the system selects the node that gives the minimum Z value defined by Eq. (8). Then, possible links originated from the selected node are considered; if the link between the selected node and the new node (n_k) also gives the minimum Z value over the other possible links from the selected node, then the link between the selected node and the new node becomes an element of the musical stream. Thus the networks are built by connecting nodes that give the locally minimum Z value.

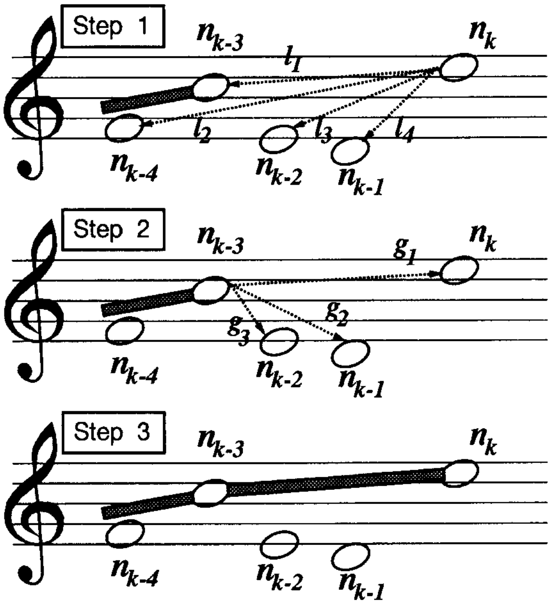

Fig. 7. A procedure for creating MSNs. Step 1. When a new node (n_k) is created, the system first chooses the link that gives the minimum Z value (l_1) among the candidate links ($l_1, \ldots, l_4$). Step 2. The system then evaluates Z values for the link candidates ($g_1, \ldots, g_3$) from the selected node (n_{k-3}), to choose the link with the minimum Z value (g_1). Step 3. If g_1 and l_1 are identical, the link composes a music stream. If a music stream from n_{k-3} was already formed in a direction other than g_1, the stream is cut; the direction of the music stream is changed to $g_1 (= l_1)$.

5. Updating probabilities based on MSNs

In this section, sound source identification with the music stream information is defined as the estimation of the a posteriori probabilities of sound sources when each note is observed.

5.1. Employment of Bayesian network

Consider a singly connected directed graph. The direction of a link corresponds to a parent-child relationship. Each node encodes a set of hypotheses and each link encodes a matrix whose elements are the conditional probabilities of child hypotheses when the parent hypotheses are true. For example, when a parent node A holds a set of hypotheses a_i $(i = 1, 2, \ldots, N)$ and a child node B has b_j $(j = 1, 2, \ldots, M)$, the link between these two nodes should include an $M \times N$ matrix whose elements are $P(b_j|a_i)$.

Here we use the term 'belief' for a dynamic conditional probability for a hypothesis held at each node, to distinguish it from the static conditional probabilities given at each link. For example, a belief vector $\mathrm{BEL}(A)$ stands for an

N-dimensional vector whose elements are the conditional probabilities for the hypotheses a_i ($i = 1, 2, \ldots, N$) maintained at node A when the belief vectors at the nodes other than A are given. The specific advantage of the Bayesian network is that each element of BEL(A) is decomposed into two components, each of which can be calculated efficiently. That is, it is shown that the BEL(A) is written as

$$\mathrm{BEL}(A) = \alpha\, \lambda(A)\, \pi(A)\,, \tag{11}$$

where α is a normalization constant, and $\lambda(A)$ and $\pi(A)$ are the N (= the number of hypotheses at node A)-dimensional vectors. The multiplication of vectors refers to an operation to obtain a vector whose elements are the products of the corresponding elements of these vectors. The main point here is that the $\lambda(A)$ and $\pi(A)$ in Eq. (11) can be recursively calculated (Kashino et al., 1995a; Pearl, 1986), which means that $\lambda(A)$ is calculated using λ vectors at the descendants and the siblings of A, and $\pi(A)$ is calculated using π vectors at the ancestors of A. Therefore the BEL vector at each node is efficiently obtained by two-path information propagation: from the children to the parents and from the parents to the children.

5.2. Information propagation on MSNs

Fig. 8 depicts how information is propagated through the MSN. In Fig. 8, the link l is just created as a fragment of the music stream by the procedure described in Section 5.1. A square stands for a belief node, which is a node holding the BEL vector, while a trapezoid denotes a data node, which is a node holding the observed data as the λ vector. That is, the λ vector at the data node is the timbre vector, each element of which is a correlation value between the agent output y_i and the corresponding part of the input waveform z.

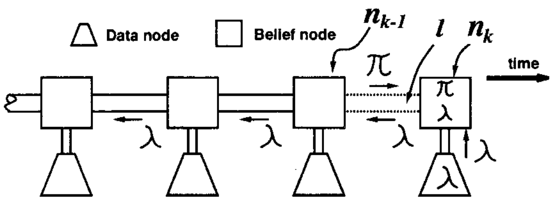

Fig. 8. Information propagation on MSN.

The λ vector at the node n_k is initially identical to the λ vector of the corresponding data node. When the link l is created, however, the π vector for node n_k is calculated using the π vector at node n_{k-1} and is propagated to n_k. Then the λ vector for node n_{k-1} is calculated using the λ vector at node n_{k-1} and propagated to n_{k-1}. The λ vector at n_{k-1} is further propagated to the parents; the propagation continues until a node that has no parents to deliver λ. For each node, a BEL vector at that moment is obtained as the 'product' of the λ vector and the π vector.

As mentioned earlier, the propagation process requires the conditional probabilities of child hypotheses when the parent hypotheses are true. In the following experiments, we simply defined the conditional probability $P(h_j|h_i)$, that is a probability of sequence of two hypotheses h_i and h_j given that h_i is true, as

$$P(h_j|h_i) = \begin{cases} \beta\left(\frac{1}{2} + c\right) & \text{if } h_j \text{ is the hypothesis for the same instrument as } h_i, \\ \beta\left(\frac{1}{2} - c\right) & \text{otherwise,} \end{cases} \tag{12}$$

where c ($0 \leqslant c \leqslant 1/2$) is a weighting coefficient and β is a normalization constant.

6. Evaluations

We have tested the proposed method using recordings of real ensemble performances listed in Table 1. These music were arranged as three-part ensembles and each part was single-pitched. Since this experiment focused on the precision of sound source identification rather than note extraction, we manually fed the system in advance with the correct pitch and time for each note.

Templates used in the adaptive template matching stage were played by different manufacturers' instruments from the ones used in the recording of the test music. We stored piano, flute, and violin templates; this means that the system presumed that each input note was played by either piano, flute, or violin. For each instrument, one template waveform is stored for each semitone over the pitch range of the instrument. The number of stored templates were, therefore, 40 for a flute (59–98 in MIDI note number), 88 for a piano (21–108), and 41 for a violin (56–96).

The number of simultaneous notes for each instrument was not given to the system. The number of parts was also unknown. The number of taps in the template filtering was 20 for those experiments where the template filtering was in action (the 'template-filtering-on' condition). The values of the parameters were chosen as listed in Table 2.

An example of the system in operation is shown in Figs. 9–11. The input here is a monaural recording of a real ensemble performance of 'Auld Lang Syne', a Scottish folk song, arranged in thee parts and performed by a violin, a flute, and a piano. Fig. 9 display the recognized music streams as well as the status of nodes for the beginning part of the song. The bars in each node indicate the belief vector at the node. The links between the nodes are the extracted music streams; it is shown that each part is correctly recognized as the music stream. The thickness of link lines corresponds to its Z value (Eq. (8)); a thick line stands for a link with the low Z value.

Figs. 10 and 11 are the score-like system outputs before and after introduction of the music streams, which demonstrate the impact of the music streams. Here all notes are displayed as quarter notes in a real time scale, because we do not introduce a note-value identification process.

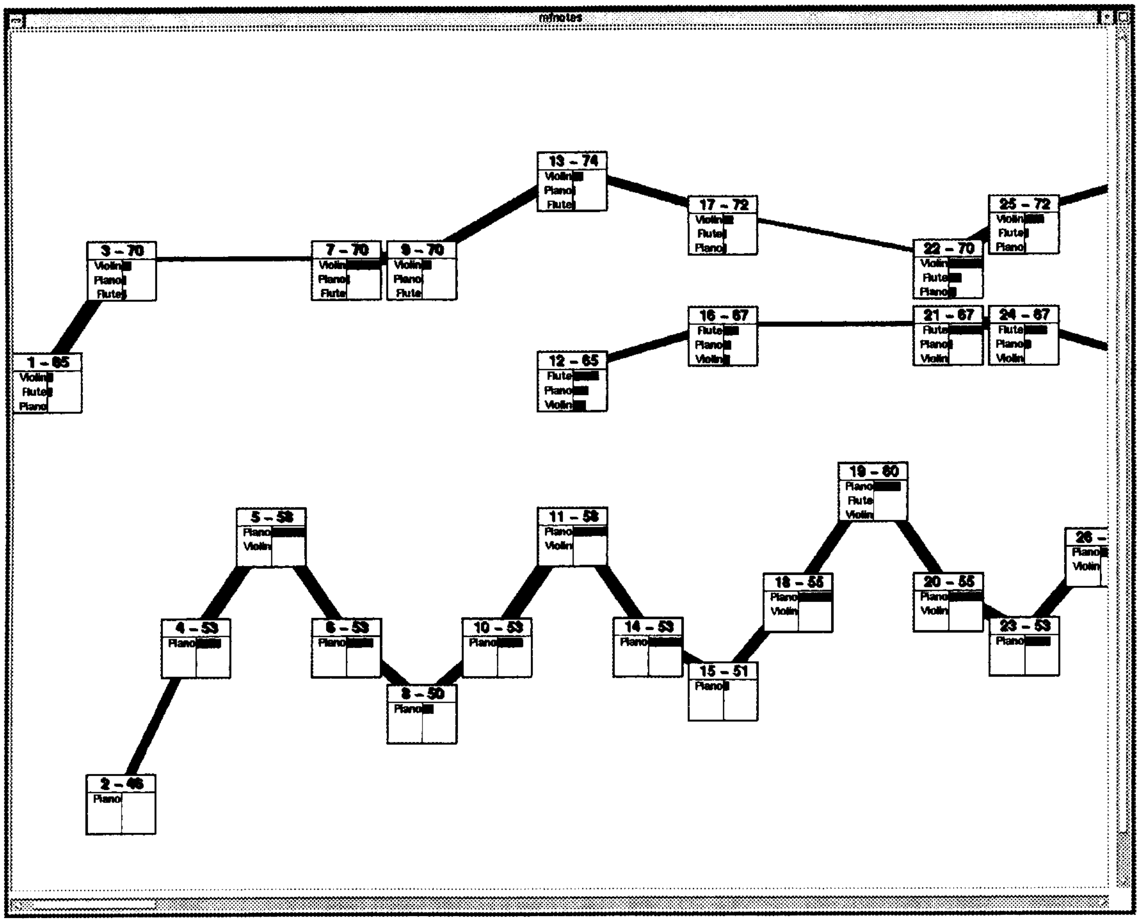
mfnotes
13 – 74 Violin Piano Flute
3 – 70 Violin Piano Flute
7 – 70 Violin Piano Flute
9 – 70 Violin Piano Flute
17 – 72 Violin Flute Piano
25 – 72 Violin Flute Piano
22 – 70 Violin Flute Piano
1 – 65 Violin Flute Piano
12 – 65 Flute Piano Violin
16 – 67 Flute Piano Violin
21 – 67 Flute Piano Violin
24 – 67 Flute Piano Violin
19 – 60 Piano Flute Violin
5 – 58 Piano Violin
11 – 58 Piano Violin
26 – Piano Violin
18 – 55 Piano Violin
20 – 55 Piano Violin
4 – 53 Piano
6 – 53 Piano
10 – 53 Piano
14 – 53 Piano
23 – 53 Piano
8 – 50 Piano
15 – 51 Piano
2 – 46 Piano

Fig. 9. Nodes after introduction of music streams. Squares stand for notes. Ordinate: pitch; abscissa: time.

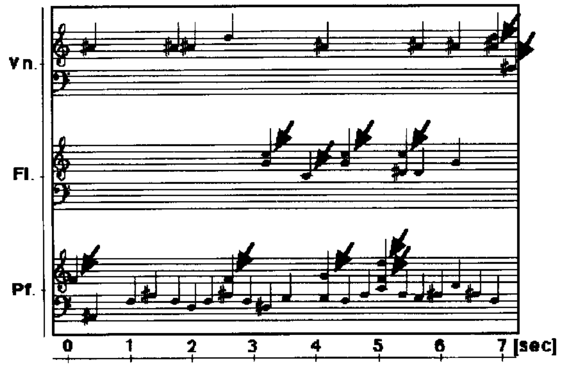

Fig. 10. Output of the system before introduction of music streams. Arrows show the incorrectly source-identified notes.

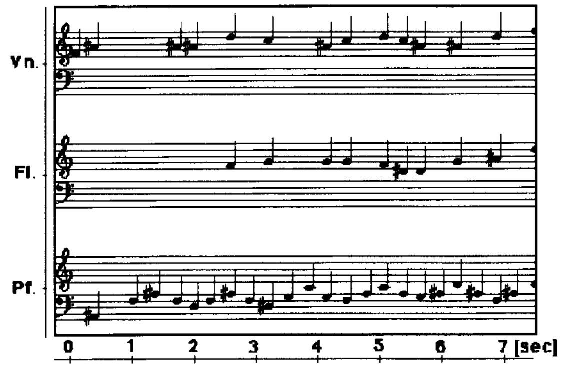

Fig. 11. Output of the system after introduction of music streams.

Comparing Fig. 11 with Fig. 10, it is observed that the several misidentified notes found in Fig. 10 are correctly modified in Fig. 11, due to the update of belief vectors as described in Section 5.

Fig. 12 shows the experimental results. The recognition rate R was simply defined as

$$R = \frac{(\#\text{correctly recognized notes})}{(\#\text{output notes in total})}. \qquad (13)$$

Here the 'template-filtering-off' condition means that the number of taps in the template filtering was chosen to be 1. Therefore turning all the elements off is equivalent to the conventional matched filtering.

It is clear that both of the adaptive template matching (PT and TF) and the musical context integration (IT, TS and RC) improves the source identification accuracy. Specifically, when all the factors discussed in Section 4 were introduced (case 11), the number of misidentification is reduced to less than half compared with the case where these three factors were not introduced (case 4). When the timbre similarity factor was solely introduced (case 5), the recognition accuracy rather deteriorated (in comparison with case 4). This is because the topology of the music stream networks did not correctly correspond to the musical part.

In this experiment, the musical role consistency (P_3) appeared to be the most effective factor. This is because the four pieces used in this experiment (Table 1), which were arranged in a traditional ensemble style, conform to the role consistency (that is, the instrument for the principal melodies or bass-lines does not change throughout the music).

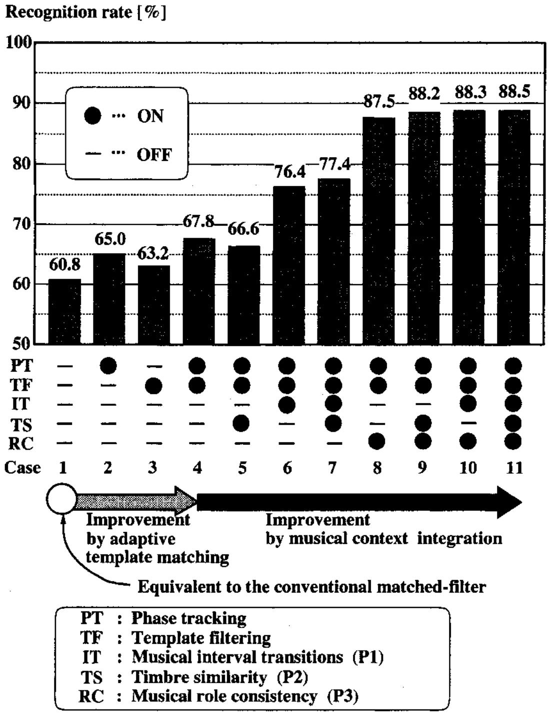

Fig. 12. The results of evaluation tests.

Table 1
Music used in the evaluation experiments

Title	Instruments (Part order)	#Notes
Annie Laurie [a]	Fl, Vn, Pf	234
Lorelei [b]	Fl, Vn, Pf	297
Dreaming of home and mother [c]	Vn, Fl, Pf	304
Auld lang syne [a]	Vn, Fl, Pf	242
Total		1077

Vn: Violin, Fl: Flute, Pf: Piano
Music by:
[a] Scotland air; [b] Friedrich Silcher; [c] J.P. Ordway.

Table 2
Values of parameters chosen in the experiments

$w_1 = 0.1$	$w_2 = 1.1$	$w_3 = 1.0$
$\tau = 2$ [s]	$a = 0.8$	$b = 0.1$
$c = 0.45$		

7. Conclusions

We have presented a new processing method of sound source identification for ensemble music. The method consists of two stages, adaptive template matching and musical context integration. Evaluation tests using recordings of real ensemble performances clearly showed that both of these techniques are effective for improving identification accuracy. Specifically, it was shown that the integration of musical context improves the precision of sound source identification from 67.8% to 88.5% on average.

However, the accuracy is not yet sufficient for the useful applications such as the automatic transcription systems and further improvement is needed. Here we have discussed the musical context in terms of a single-pitched melody; however, the chord will be also an important musical context factor. Thus an extension of the music stream network method to include chord transitions will be considered in future work. We are also planning to evaluate the system using musical performances that have further varieties; for example, performances that include musical instruments different to those reported here, and also performances with more than three parts.

Acknowledgements

We would like to thank Hiroshi G. Okuno, Takeshi Kawabata, and Tomohiro Nakatani for their stimulating discussions. We also wish to thank Dr. Youichi Tohkura and Dr. Ken'ichiro Ishii for their support.

References

Bell, A.J., Sejnowski, T.J., 1995. An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7, 1129–1159.

Bregman, A.S., 1990. Auditory Scene Analysis, MIT Press, Cambridge, MA.

Brown, G.J., Cooke, M., 1994. Perceptual grouping of musical sounds: A computational model. J. New Music Research 23 (1), 107–132.

Cardoso, J.F., 1989. Source separation using higher order moments. In: Proceedings of ICASSP-89, pp. 2109–2112.

Chafe, C., Jaffe, D., 1986. Source separation and note identification in polyphonic music. In: Proceedings of ICASSP-86, pp. 1289–1292.

Cooke, M., 1991. Modelling auditory processing and organisation. Department of Computer Science, University of Sheffield, Ph.D. Thesis.

Cosi, P., Poli, G., Lauzzana, G., 1994. Auditory modelling and self-organizing neural networks for timbre classification. J. New Music Research 23 (1), 71–98.

Ellis, D., 1996. Prediction-driven computational auditory scene analysis. Ph.D. Thesis, Deptartment of Electrical Engineering and Computer Science, M.I.T.

Flanagan, J.L., Johnston, J.D., Zahn, R., Elko, G.W., 1985. Computer-steered microphone arrays for sound transduction in large room. J. Acoust. Soc. Amer. 78 (5), 1508–1516.

Kashino, K., Tanaka, H., 1993. A sound source separation system with the ability of automatic tone modeling. In: Proceedings of the International Computer Music Conference, pp. 248–255.

Kashino, K., Nakadai, K., Kinoshita, T., Tanaka, H., 1995a. Organization of hierarchical perceptual sounds. In: Proceedings of the International Joint Conference on Artificial Intelligence, Vol. 1, pp. 158–164.

Kashino, K., Nakadai, K., Kinoshita, T., Tanaka, H., 1995b. Application of Bayesian probability network to music scene analysis. In: Working Notes of the Computational Auditory Scene Analysis Workshop, IJCAI-95, pp. 32–40.

Katayose, H., Inokuchi, S., 1989. An intelligent transcription system. In: Proceedings of International Conference on Music Perception and Cognition, pp. 95–98.

Lee, T., Orglmeister, R., 1997. A contextual blind separation of delayed and convolved sources. In: Proceedings of ICASSP-97, pp. 1199–1202.

Lesser, V., Nawab, H., Gallastegi, I., Klassner, F., 1993. IPUS: An architecture for integrated signal processing and signal interpretation in complex environments. In: Proceedings of the 11th National Conference on Artificial Intelligence, pp. 249–255.

Maes, P. (Ed.), 1990. Designing Autonomous Agents, MIT Press, Cambridge, MA.

Mitchell, O.M.E., Ross, C.A., Yates, G.H., 1971. Signal processing for a cocktail party effect. J. Acoust. Soc. Amer. 50 (2), 656–660.

Mont-Reynaud, B., 1985. Problem-solving strategies in a music transcription system. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 916–918.

Nakatani, T., Okuno, H.G., Kawabata, T., 1995. Residue-driven architecture for computational auditory scene analysis. In: Proceedings of the International Joint Conference on Artificial Intelligence, Vol. 1, pp. 165–172.

Nehorai, A., Porat, B., 1986. Adaptive comb filtering for harmonic signal enhancement. IEEE Trans. on ASSP 34 (5), 1124–1138.

Niihara, T., Inokuchi, S., 1986. Transcription of sung song. In: Proceedings of ICASSP-86, pp. 1277–1280.

Parsons, T.W., 1976. Separation of speech from interfering speech by means of harmonic selection. J. Acoust. Soc. Amer. 60 (4), 911–918.

Pearl, J., 1986. Fusion, propagation, and structuring in belief networks. Artificial Intelligence 29 (3), 241–288.

Piszczalski, M., Galler, B.A., 1977. Automatic Music Transcription. Computer Music Journal 1 (4), 24–31.