Journal of Information Processing Vol.21 No.2 274–282 (Apr. 2013)

[DOI: 10.2197/ipsjjip.21.274]

Recommended Paper

Possessing Drums: An Interface of Musical Instruments that Assigns Arbitrary Timbres to Personal Belongings

Kazuhiko Yamamoto[1,a)]

Received: June 27, 2012, Accepted: January 11, 2013

Abstract: In this work, I propose an interface for musical instruments for assigning arbitrary timbres to arbitrary objects including personal belongings such as a table or cup, or actions such as vocalization by audio signal processing, to enable the users to play music as if they were playing the actual acoustic musical instrument which generates the simulated timbres. This system requires no special device, only a standard microphone. The assigned timbres are produced not by a triggered PCM (pulse-code modulation) waveform in response to the detected attacks in the microphone input source but by the modeling process of the system that generates the timbres by modifying the microphone input source itself. It thereby enables the users to play music with very sensitive expression, including very small sounds, fast passages, and the effects of playing style. Additionally, in this system, we can assign separate individual timbres to each of a set of objects at a time and play polyphonic music.

Keywords: human-computer interface, digital musical instrument, audio signal processing, sound synthesis

1. Introduction

The experience of playing musical instruments is very attractive to all people. However, the set up and transportation of the physical instruments are not easy, especially for acoustic instruments, and the maintenance is often troublesome. For these reasons, playing actual musical instruments can be a very expensive undertaking. On the other hand, there are many portable digital musical instruments, and musical instrument applications on portable devices such as smart phones. Through the use of them, we can solve the problems of portability and the difficulties of the maintenance of actual acoustic musical instruments. However, in most cases, these conventional portable musical instuments employ touch panels or mechanical buttons for user input interface. These input interfaces have several fatal problems for the interface of musical instrument: large latency to response, limited response however the user plays, no tactile sensation, and quite limited space for playing.

Meanwhile, whether we have an ability to play musical instruments or not, we often imitate musical instruments by extemporizing a rhythm, beating physical objects such as a table in our vicinity by hand like drums, or singing like playing a saxophone. Look at in this way, it can be said that all everyday objects around us are the most familiar musical instruments. Just as with an acoustic musical instrument, the sound we generate with commom objects dynamically and sensitively changes depending on how we use them. In other words, it means that all of our movements and how to play affect the output sound and become meaningful input. In this respect, every physical object around us can have as much expressiveness as actual acoustic musical instruments. However, it is hard to say that these timbres are attractive in comparison with that of actual musical instruments.

If these acts that imitate musical instruments could be augmented with the aid of the computer and we could perform music in arbitrary timbre using common objects while keeping the advantage of the physically dynamic changing sound of them, we could always use everyday objects around us exactly as actual acoustic musical instruments and play music easily everywhere and anywhere without discrimination. For example, regardless of whether the person is an amateur or professional player of musical instrument, child or adult, they could enjoy air-drum playing with various timbres which are dynamically changing in response to the player's techniques. They could also begin the session together at once even in a car and enjoy a musical experiment. This would also have tremendous values for the field of music education for children at an early age because every object that is touched becomes a musical instrument and the hurdles in the path of using musical instruments would be alleviated. On the other hand, for professional musicians, they could use not only their usual musical instrument but also every object around them as a musical instrument and combine them for live performances. They could also augment even their usual instrument itself with the aid of a computer and mix this with the original sound for a new mode of expression. Moreover, it can be expected that there will be some applications for which such a computer interface will enable highly intuitive feedback. For example, intuitive feedback for the process of designing a musical instrument by using everyday objects (such as a desktop) as input devices is possible if the computer could output the tone corresponding to the 3D ge-

[1] YAMAHA Corporation, Hamamatsu, Shizuoka 430–8650, Japan
[a)] yamotulp@gmail.com

The content of this paper was reported at Interaction 2012 in March 2012 and the paper was recommended to submit for Journal of Information Processing society.

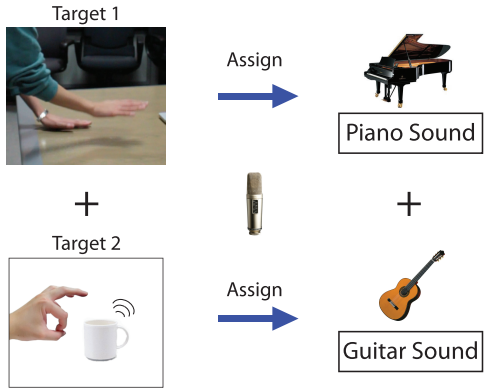

Fig. 1 Assigning the user's favorite timbres to multiple common objects. For example, when the user hits the table, it outputs a piano tone. Additionally, when the user hits the table and flicks the cup at the same time, they output a piano tone and guitar tone simultaneously.

ometry of the musical instrument which the user is designing in computer graphics.

Therefore, in this paper, I propose an interface for digital musical instruments, called "Possessing Drums," which uses an audio signal processing to assign arbitrary timbres to common objects in the user's environment. Only a microphone is needed as the input sensor for this system; no other special devices are required. In this system, any objects and actions that produce a sound can be an assignable target. For example, a user can assign a piano timbre to a table (the target) using this interface, and the user can play the piano timbre by striking the table with the user's hand such that the table effectively becomes a piano (**Fig. 1**). My approach also can assign multiple timbres to each of multiple targets simultaneously by real-time sound source separation and can play polyphonic music in spite of using one microphone only (Fig. 1).

In addition, what is assigned to the target are the characteristics itself of the assigned timbre, as determined by the nature of the sound generation process in physical objects in the real world. For example, in the case of the sound generated when the table is struck with the user's hand, the sound generation process encompasses the fact that the struck table vibrates, the sound is emitted to the air through the table surface, and finally that sound reaches the user's ears. Possessing Drums simulates these characteristics, and can utilize the advantages of the approach that treats the sound itself from the microphone as user input because this system adds the assigned characteristics itself to the user input sound by audio signal processing. So, this system produces dynamically changing live sound as if the user is directly touching the actual object that produces the assigned timbre.

My system even works not only on a desktop PC but also on common portable devices such as a smart phone. We can expect that most mobile devices have at-least one or more microphones which is the only required hardware and have sufficient computational specifications for processing the proposed algorithm.

2. Related Work

Past interfaces for playing musical instruments in real-time can be broadly classified into two categories from the standpoint of the sensing method. The first is the "Trigger Method," which uses the amount of change of any of the sensors as a trigger signal for voicing. Including the common electronic musical keyboard, most conventional digital musical instruments adopting the MIDI protocol [4] for voicing, and most of the past interfaces proposed for musical instruments belong to this category. To define the amount of change as a trigger signal, the definition of the threshold for what amount of change generates a trigger signal is needed (the first row of **Fig. 2**). The existence of these thresholds means that it is impossible to react to small changes of sensor input that are below the defined threshold (the second row of Fig. 2). In addition, once an input sensor records a signal change that exceeds the threshold, it is impossible to detect the next trigger until the amount of change of the sensor falls below the threshold again (the third and forth row of Fig. 2). So, this method is limited in how quickly it can react to changes in input. This is why this method has a limitation in terms of sensing the sensitive motions of the performer, making it impossible to react to very fast passages and very small amplitude inputs. This limitation seriously undermines our sensation of touching the musical instrument itself while playing. Thus, because this method cannot utilize the dynamically changing sound of everyday objects around us, which is the purpose of this paper, it is not useful for this study. The second approach is to process the sound from a microphone or input data from any sensor itself into output such as effects unit [9], as is done by the KORG WaveDrum [6]. In this approach, we can make an expression as well as an actual acoustic musical instrument because it is possible to capture all user input from the sensor as meaningful input for the system, no matter how delicate, fast, or small. In a broad sense, Theremin [8] also falls into this category, in that it continuously translates its change in electric capacity into sound. The advantage of directly processing the input data itself to generate output sound is the capability of reacting to all of a performer's movements. Because of this advantage, this approach can emphasize our feeling of touching the playing musical instrument itself. Therefore, I have applied this approach.

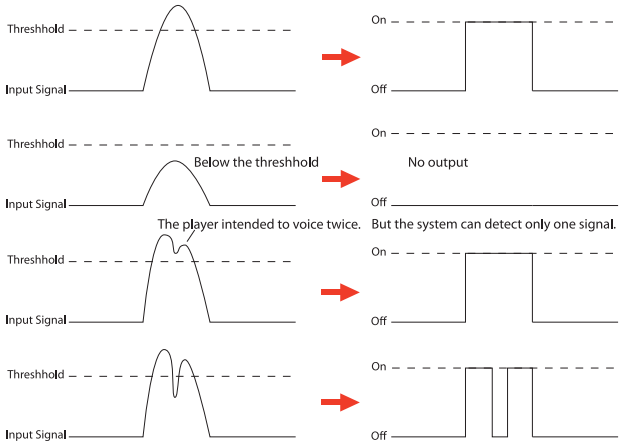

Fig. 2 The trigger method.

On the other hand, several past approaches that can treat personal belongings as musical instruments have been proposed. "The Sound of Touch" [10] is a wand-shaped device that contains an embedded microphone and a piezo vibration sensor. This device records the vibration of physical objects manipulated by

the user through brushing, scraping, and striking motions, and generates the output sound by convoluting the recorded vibration with the impulse response simulating the user-assigned timbre. This system enables one to play music with arbitrary timbre using whatever objects are in reach by utilizing the vibrations from the sensor itself. However, it is not so much that this approach assigns a timbre to objects, but rather the device itself simply is a stand-alone musical instrument. So, this approach is unfit for the purpose of this paper, to use everyday objects as musical instruments without any special devices. The KORG WaveDrum mini [7], which uses a clip-shaped piezo vibration sensor, can treat the vibration of a clipped object as input to the WaveDrum described above. This approach has an advantage in robustness in that it can definitely capture the target's vibration. However, it is impossible to adopt a target that cannot be clipped like phonation, and using this approach may affect the vibration of the target object because the vibration of the object is damped by clipping. A third approach called "TableDrum" [5] generates the PCM wave sound assigned to the sound from objects like the sound of a table struck by the user. TableDrum treats any momentary increase of the microphone signal amplitude over the defined threshold as a trigger to begin to produce sound. This is consistent with the objective of this paper. However, because this sensing method is the "Trigger Method" as described above, TableDrum cannot fully utilize the advantage of selecting the sound itself from the microphone as the input method. Moreover, TableDrum has the limitation that the output sound can only be monophonic because it can detect only one target at a time.

3. Approach

Figure 3 shows the outline of my approach. This system takes the sound from a microphone as input, and outputs the assigned sound to a loudspeaker. The user registers the sounds that they want to be assigned to the target and the sounds they want to assign; the user also sets which timbre is assigned to which target in advance. When the user causes one of the targets to make a sound, the system detects which registered target is the input and outputs the corresponding timbre. If the user makes multiple target sounds simultaneously, the system separates each target signal from the mixed signal containing multiple target sounds, and outputs the timbres associated with each of the targets.

In this paper, to synthesize the assigned timbre, I apply an approach to simulate the characteristics of the physical sound generation process of the assigned timbre instead of simply playing a PCM wave sound. The process of producing all sounds in the real world can be divided into two parts, driving and propagation (**Fig. 4**). For instance, in the case of the sound produced by a hand striking a table, the striking with the hand is the driving part, and the process by which the vibration propagates through the table and is emitted to the air as a sound wave is the propagation part. Another example is phonation: Here, the vibration of vocal folds is the driving part, and the emission from the mouth after passing through the vocal tract is the propagating part. Possessing Drums assigns a timbre by replacing only the propagating part of the target sound in this process with the propagating part of the assigned timbre (**Fig. 5**). For example, when the user assigns a drum timbre to the sound of striking a table with a hand, the combination of the propagating part of the drum sound with the driving part of the target sound (which results from striking the table with the hand) is obtained. As a result, the user can experience the sensation of playing the drum with his hands. Because what is replaced is only the propagating part, if the user scratches

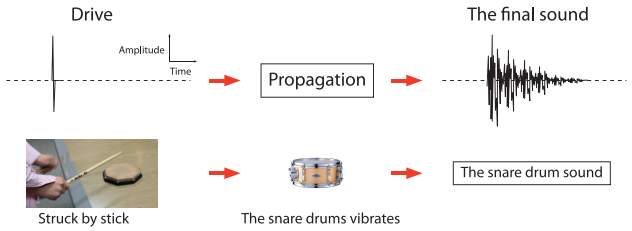

Fig. 4 The physical sound generation process can be devided into two parts, the driving part and the propagating part.

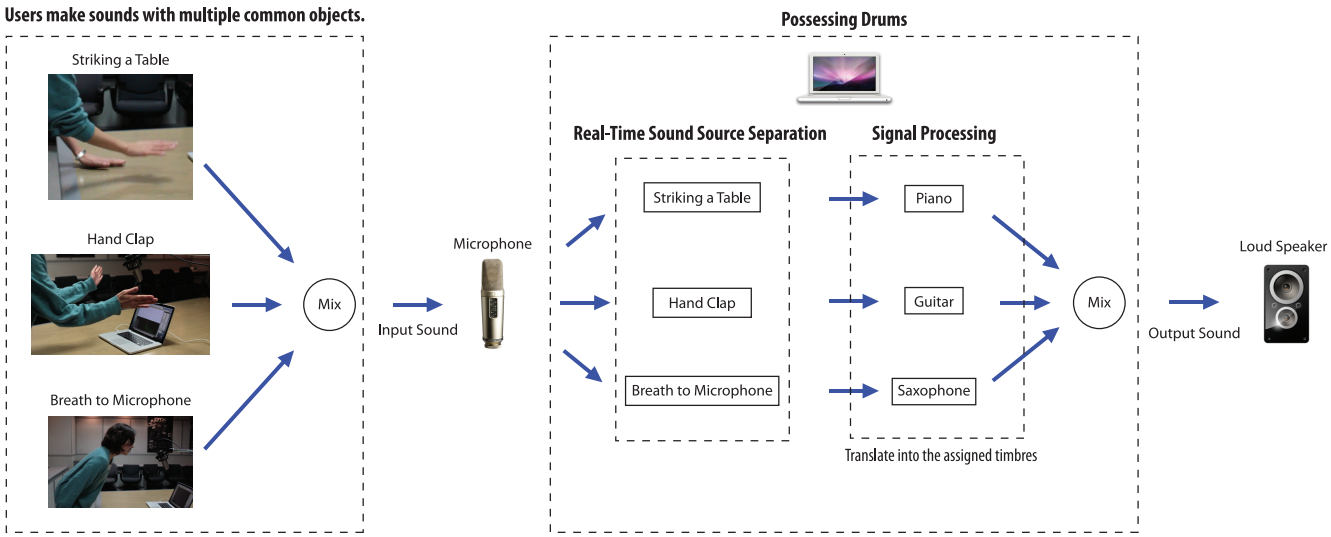

Fig. 3 The outline of possessing drums. First, the sound from the microphone is separated into each target. Second, each separated sound is modified by signal processing to translate the timbre into the assigned timbre.

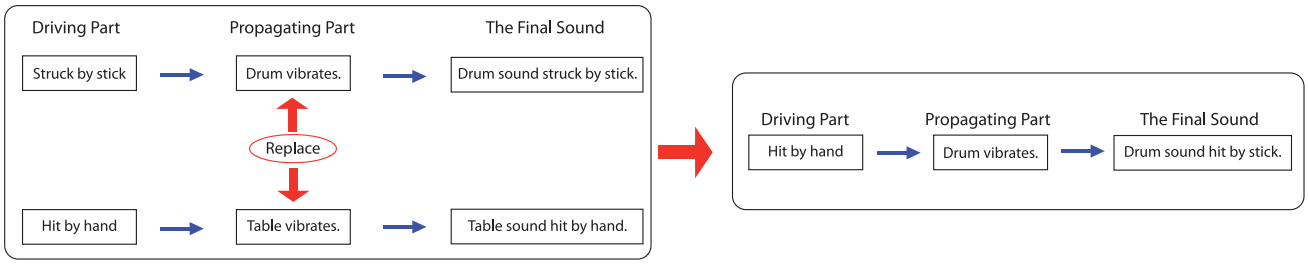

Fig. 5 This interface replaces the propagating part of the target with that of the assigned timbre.

the table, it is equivalent to scratching the drum. In addition, anything that makes sound can be the target in this system; if the user registers the sound of breathing to a microphone as the target, the user can drive the drum timbre by breathing, even though it is impossible to sufficiently excite a drum surface by breathing in real life.

4. Hardware

This system needs only a microphone as sensor and a loud speaker as hardware equipment without any special devices except a computer for audio signal processing. So, it is very easy to set up.

5. Software

This system has two modes, "Register Mode" and "Performance Mode." In "Register Mode," the user registers the timbres he wants to assign and the targets he wants to be assigned respectively. On the other hand, in "Performance Mode," the user actually plays this system. "Performance Mode" consists of three blocks, "Recognition Block," "Quasi-Inverse Filter Block," "Synthesis Block." The "Recognition Block" separates the input sound from the microphone to sound source and detects each assigned target. "Quasi-Filter Block" extracts the driving part of each target sound. "Synthesis Block" adds the propagation part which simulates the assigned timbre. The details are described as follow.

5.1 Register Mode

In this system, first, it is necessary to register the information of the timbre wants to assign and the target sound wants to be assigned by recording each sound in advance. In a linear system, the vibration of an object $y(t)$ can be expressed by the sum of decay sines, called vibration modes.

$$y(t) = \sum_{n}^{N} A_n e^{-d_n t} sin(2\pi f_n t) \tag{1}$$

where A_n, d_n, f_n is the amplitude, damping coefficient and frequency of the n-th mode respectively. These parameters are called "modal parameters." "Register Mode" estimates these parameters from the recorded sound, and registers to this system. The modal parameters are computed after splitting the recorded sound into one shots by detecting its onsets. There are several methods to estimate the modal parameters from a not ideal impulse response. Here, I apply *Sirdly*'s method [1] which uses the combination of ESPRIT method with gabor transform. In addition to these modal parameters, this system retains the average power spectrum while sounding. The average power spectrum is used for the template data for sound source separation of microphone input in "Recognition Block" that is described later. However, it is not necessary to retain the sound wave data itself.

5.2 Performance Mode

In performance mode, this system detects the target sounds from a microphone input, and synthesizes the assigned timbre as output sound.

5.2.1 Recognition Block

The sound input from the microphone is mixtured sound containing multiple target sounds, no related noise, and feedback of this system itself from a loud speaker. So, it is necessary to separate and extract each target sound from the mixtured input. In addition, to remove the feedback sound of this system itself is important to prevent the howling because this system processes the sound itself from the microphone for output. Here, the average power spectrums of the registered target sound and output of this system is already known in the input mixtured sound. In this study, I use this average power spectrums as templates for the basis, and apply sound source separation partly with templates by modified β-Divergence NMF (Non-Negative Matrix Factorization) [3]. In NMF, spectrogram matrix V is expressed as follows on the assumption of the sparsity,

$$\mathbf{V} \approx \mathbf{WH} \tag{2}$$

where $\mathbf{W}$ denotes the dictionary matrix that expresses the average power spectrums of each timbre contained in $\mathbf{V}$. $\mathbf{H}$ denotes the activation matrix which explains each timbre's amplitudes at each time. NMF separates the input spectrogram matrix into the multiplying of the dictionary matrix with the activation matrix. β-Divergence NMF performs this matrix factorization by minimizing β-Divergence D_β $(\mathbf{V}|\mathbf{WH})$ between $\mathbf{V}$ and $\mathbf{WH}$ subject to the combination of $\mathbf{W}$ and $\mathbf{H}$. To solve this problem, $\mathbf{V}$ and $\mathbf{H}$ are updated iteratively until convergence as follows.

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^{\mathbf{T}}((\mathbf{WH})^{\beta-2} \cdot \mathbf{V})}{\mathbf{W}^{\mathbf{T}}(\mathbf{WH})^{\beta-1}} \tag{3}$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{((\mathbf{WH})^{\beta-2} \cdot \mathbf{V})\mathbf{H}^{\mathbf{T}}}{(\mathbf{WH})^{\beta-1}\mathbf{H}^{\mathbf{T}}} \tag{4}$$

In each iteration, the basis vectors in the dictionary matrix has a template that doesn't need to be updated. On the other hand, the basis vector in the dictionary matrix which corresponds to no related noise needs to be updated because it has no template data. In general, sound source separation using NMF is a non-realtime

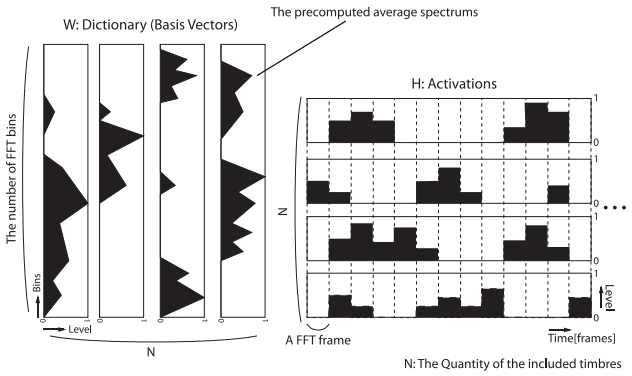

Fig. 6 Dictionay matrix and activation matrix.

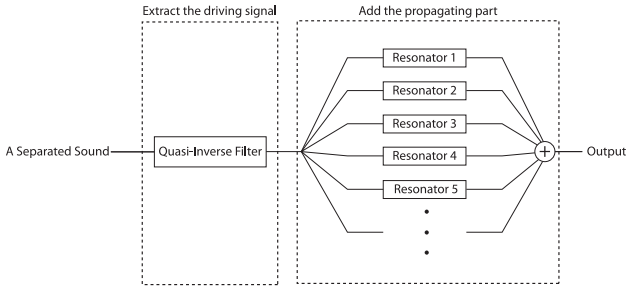

Fig. 7 Quasi-inverse filter block and synthesis block.

operation because the time length of spectrogram **V** is too long in comparison to the window size of FFT. In this paper, to operate sound source separation in realtime, I compute this factorization at the current frame of FFT only (**Fig. 6**). Thus, the activation matrix becomes a column vector of its size which is equal to the number of registered targets in advance plus two for no related noise and feedback noise to prevent howling. Moreover, I can accelarate the convergence and decrease the number of iterations by setting the result of the previous frame to **H** as the initial value of the current frame, because the input signal is continuous. In addition, to prevent howling, passing through the LMS adaptive filter [11] before processing this sound source separation is also useful.

5.2.2 Quasi-Inverse Filter Block

Each separated sound in the previous block already contains both the driving part and the propagating part. So, it is necessary to extract the driving part as far as possible only from this sound. To obtain the driving part signal, I compute the quasi-inverse filter processing to the sound (**Fig. 7**). The quasi-inverse filter can be approximated by the mode numbers of serial connections of ByQuad filters from precomputed modal parameters in the register mode. Eq. (5) describes the transfer function of the 2nd-order ByQuad filter that has a dip in frequency f_z.

$$\frac{Y}{X} = \frac{G(1 - 2r_z \cos(2\pi f_z T)Z^{-1} + r_z^2 Z^{-2})}{1 - 2r_p \cos(2\pi f_p T)Z^{-1} + r_p^2 Z^{-2}} \tag{5}$$

where G denotes the gain, r_z denotes the damping coefficient at zero, r_p and f_p indicate the frequency and the damping coefficient at pole respectively.

5.2.3 Synthesis Block

This block inputs the driving signal extracted by the quasi-inverse filter into parallel connected resonators by an amount equal to the number of the modes that simulate the propagating part of the assigned timbre (Fig. 7). There are many types of resonator structures that have been used to simulate sounding objects. In this paper, I apply the modal resonator proposed by Kees van den Doel et al. [2], which consists of parallel banks of second order resonant filters, each with individual coupling constants and damping. Now we define the output of the resonator as the angular frequency of n-th mode ω_n as $y_n(m) = u(m) + iv(m)$, then $u(m)$ and $v(m)$ are expressed as follows.

$$\begin{aligned} u(m) &= c_r u(m-1) - c_i v(m-1) + a_n F(m), \\ v(m) &= c_i u(m-1) + c_r v(m-1) \\ c_r &= e^{-d_n/S_R} \cos(\omega_n/S_R), \\ c_i &= e^{-d_n/S_R} \sin(\omega_n/S_R), \end{aligned} \tag{6}$$

where a_n, d_n, S_R, $F(m)$ denote the amplitude of n-th mode, damping coefficient, sampling rate, external force respectively. Because $y_n(m)$ is complex amplitude, I output $Im(y_n(m))$ as the final sound.

6. Evaluation

In this section, I examine the proposed method.

6.1 Real-time Sound Source Separation

First, I tested the performance of real-time sound source separation. I compared the accuracy of the proposed algorithm with normal β-divergence NMF with template. The algorithms to separate out sound sources from single-channel audio mixtures were implemented in C++. In normal NMF, I fixed the dictionary matrix and did not update it. I evaluated the performance of these algorithms using three quality measures: the signal-to-distortion ratio (SDR), the signal-to-interference ratio (SIR), and the signal-to-artifact ratio (SAR). These measures are widely used for the evaluation of separation quality [12]. Here, I decompose each estimated source s_{final} into four terms.

$$s_{final} = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{7}$$

where s_{target} is a version of the original source modified by an allowed distortion, and where e_{interf}, e_{noise} and e_{artif} are respectively the interferences, noise and artifacts error terms [12]. Using these terms, each measure is defined by

$$SDR = 10\log_{10} \frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2}, \tag{8}$$

$$SIR = 10\log_{10} \frac{\|s_{target}\|^2}{\|e_{interf}\|^2}, \tag{9}$$

$$SAR = 10\log_{10} \frac{\|s_{target} + e_{interf} + e_{noise}\|^2}{\|e_{artif}\|^2}. \tag{10}$$

For the input signal, I prepared a MIDI file containing 6 timbres (1:piano A4, 2:piano E3, 3:piano D5, 4:guitar B3, 5:hi-hat cymbal, and 6:kick drum) and rendered it as a WAVE file. The WAVE file is 5.0 [s] in length with a sampling frequency of 44.1 [kHz]. I used the precomputed average power spectrums of individual notes contained in the test MIDI file for the template of basis

Table 1 Mean SDR, SIR and SAR for separated sound sources[dB].

	SDR	SIR	SAR
Normal NMF	6.73	2.22	4.57
Real-time NMF	6.71	2.21	4.53

vectors for the dictionary matrix (the top of **Fig. 8**). The bottom of Fig. 8 shows the piano roll of this MIDI file which indicates the sounding interval of each timbre. In the piano roll of Fig. 8, the orange bar indicates that the corresponding timbre is sounding. The numbers below the top figure of Fig. 8 and the left side of the other three figures indicate the kind of timbre. The second and third rows of Fig. 8 show the result of the activations obtained by each of the respective methods, and **Table 1** shows the SDR, SIR, and SAR, all in [dB]. The window size of FFT was 256 samples, and the sampling frequency was 44.1 [kHz]. Naturally, the accuracy of the proposed method is slightly inferior to normal NMF. However the performance of the proposed method does not decrease considerably, remaining at the accuracy that is rarely different from normal NMF. So it can be said that sufficient accuracy is achieved for this case. Under these conditions, the proposed algorithm takes about 1.0 [ms] per frame on average; when I set the window size to be 256 samples with a sampling frequency of 44.1 [kHz], this is about 6 [ms]. Therefore, it can be verified that real-time operation of this algorithm is possible.

Generally in NMF, every input sound is clustered into the closest of the sounds registered in advance. So it can be said that, if the user registers only one sound of the table as a trigger, the difference in what is used for making sound, by the user's hand or by a stick, is recognized as the result of a different playing technique of a target. If the user registers each sound of several playing techniques, these sounds are recognized as individual targets. This feature has the advantage that the user can control the expressiveness by selecting whether to include the difference of playing technique as a different target.

6.2 Method for Assigning Timbre

I verified the validity of the proposed method for assigning an arbitrary timbre to other sounds. First, I tested the algorithm using a piano tone by reconstructing the original signal. I attempted to reconstruct the original signal by inputting the extracted driving signal from a piano sound to the propagating part estimated from the same piano tone. If the reconstructed sound was very similar to the original sound, it could be said the proposed method performed well. The spectrogram of this original piano A4 tone is shown in the top part of **Fig. 9**. The driving part of a piano tone involves striking the piano string with a felt hammer. The result of extracting this driving signal from the original piano A4 tone by the proposed method is shown in the middle part of Fig. 9. The result is a pulsive signal similar to the signal produced by striking a hard material with a felt hammer. To reconstruct the original piano tone, I input this extracted driving signal into the synthesis block, adding the propagation part estimated from the original piano A4 tone. The bottom part of Fig. 9 shows the spectrogram of the result. This result is very similar to the original piano tone. We can find the same modes are excited in these two spectrograms. In this result, the higher harmonic overtones are less than that of the original. This is because there is a limitation

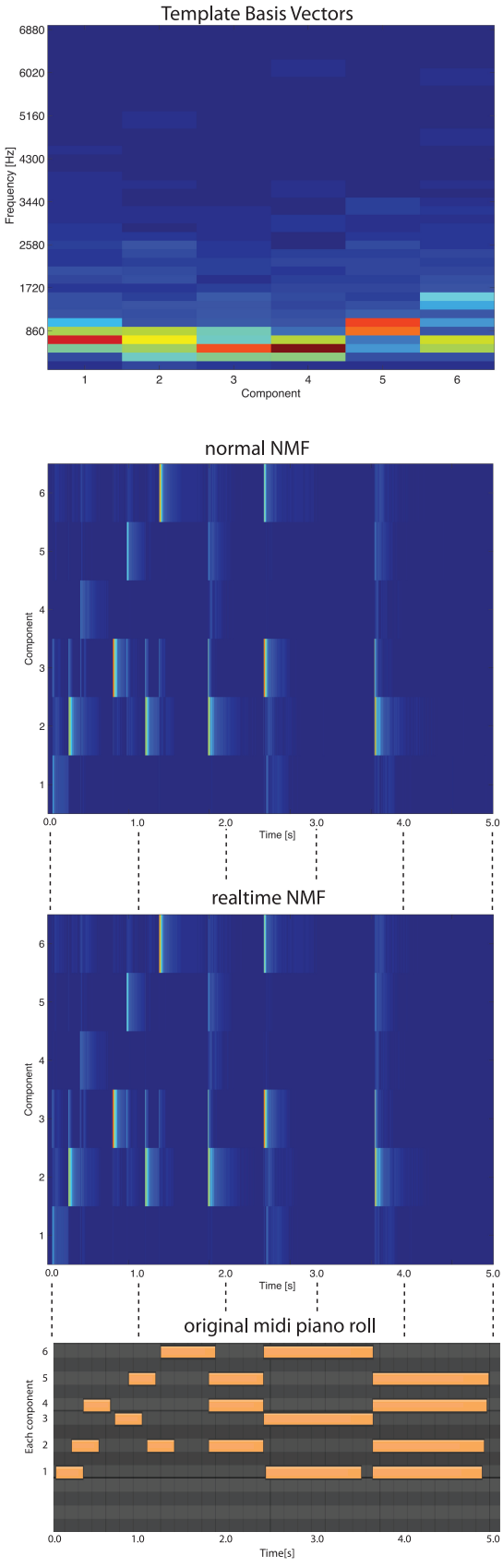

Fig. 8 The result of the sound source separation test.
Top: The dictionary matrix used for template
Second: The activation matrix of Normal NMF
Third: The activation matrix of Real-time NMF
Bottom: The MIDI piano roll of the input source. The orange bar indicates that the corresponding timbre is sounding.

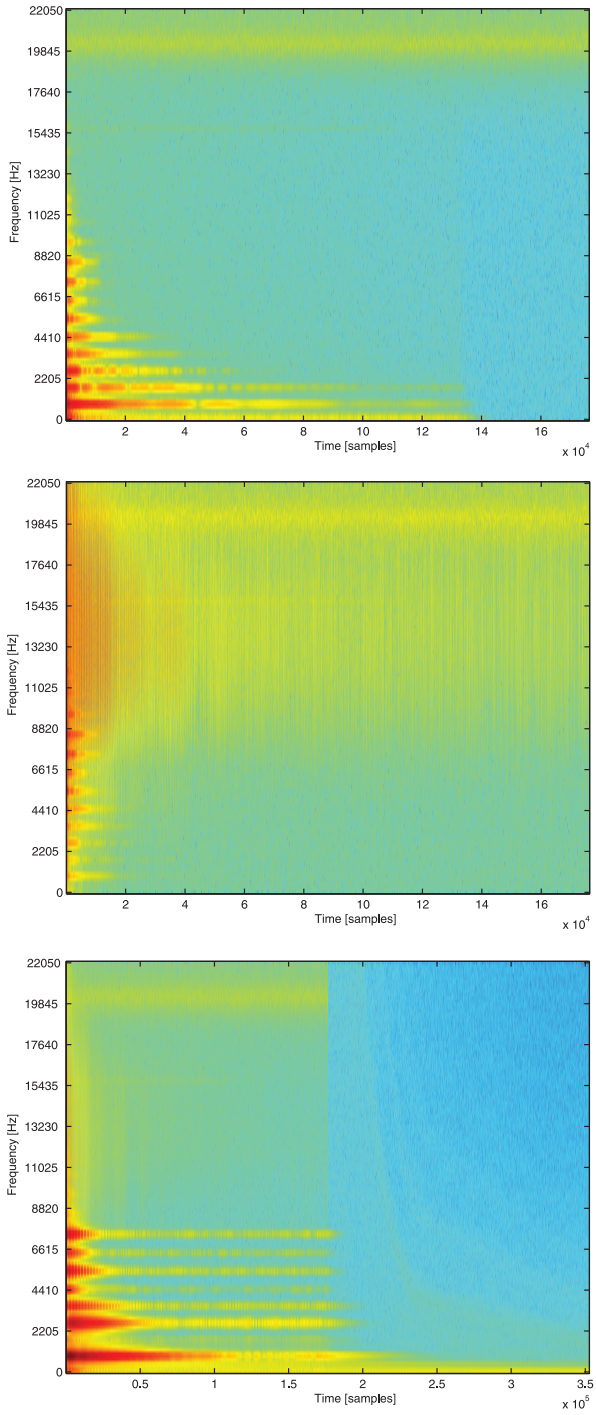

Fig. 9 Reconstruction of the original piano A4 tone by the proposed method.
Top: The original piano tone
Middle: The extracted driving signal
Bottom: The reconstructed result.

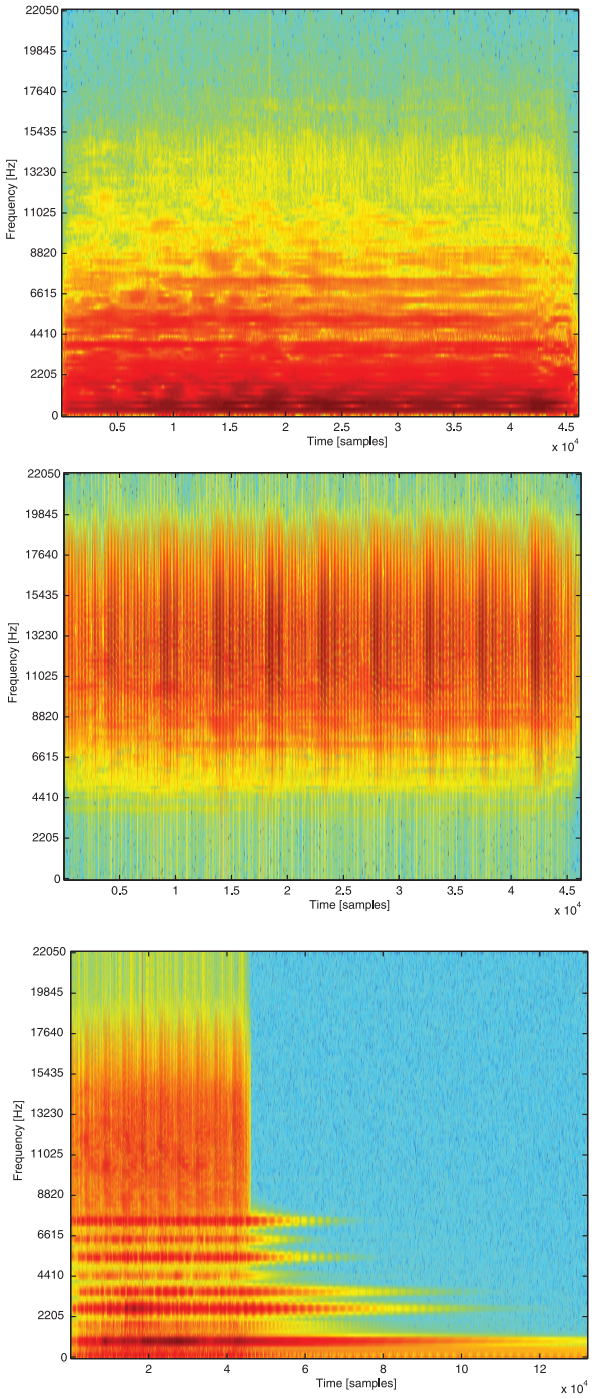

Fig. 10 Assigning a piano A4 timbre to a violin long tone with vibrato.
Top: The violin E6 long tone with vibrato (the target)
Middle: The extracted driving signal from the target.
Bottom: The synthesized result.

to the number of modes that can be simulated in this system.

Another experiment was carried out. I assigned the piano A4 timbre of the previous experiment to a violin sound (target). The driving part of a bowed string instrument such as a violin is expected to be a very fast sequence of pulses owing to the periodic motion as the string is repeatedly pulled with bowing and returned to its original position when the pulled force exceeds the limit of the friction between the bow and the string. The spectrogram of the extracted driving signal from a violin E5 long tone with vibrato (the original spectrogram is shown in top of **Fig. 10**) is shown in the middle part of Fig. 10. We can find that the very fast sequences of pulsive signal by removing the harmonic overtones is obtained in the result. In addition, this result has no pitch tone. Next, I input this driving signal to the synthesis block, adding the propagating part of the piano A4 timbre in the same way as in the previous experiment. The result is the bottom part of Fig. 10. In this result, modes the same as the piano A4 sound are excited, and a piano-like long fluctuated sound as if it is played by bowing is obtained. From this result, the algorithm for assigning an arbitrary timbre to other sounds is verified.

Fig. 11 The play scene with possessing drums. The user plays a piano tone by striking the table with her hand.

6.3 Playability

I examined the playability of the entire system by having some users actually play music with this system (**Fig. 11**). The players assigned their favorite timbres to objects or actions like a table, a coffee mug, or breathing into the microphone. Similar to actual acoustic musical instruments, what objects are used and how the user makes sounds fully affect the output sound. Because this system processes the input sound itself from the microphone, it can capture fast passages and small inputs and then include these effects in the output. Therefore, this system can capture the sensitive expression of the performers. Additionally, this system also enables the users to perform with expression impossible in reality, like playing a piano timbre by brush or a guitar by voice. As a result, the users played music expressively using this interface, and I verified the large potential of the proposed interface as a musical instrument. In this experiment, the buffer size of the audio device was set to 256 samples with a sampling frequency of 44.1 [kHz]. Under these conditions, the latency until voicing is about 6.0 [ms]. There was no difficulty in performing music in real-time with this latency. However, in this experiment, a recognition error problem was observed, resulting from the treatment of unwanted noise as signal accounting for the lack of accuracy. In addition, when the user selects ambiguous sounds like phonation as a target, accurately recreating the target sound itself may be difficult for the user. This causes a problem in obtaining the intended result of the user. Addressing this problem is left to future work.

7. Conclusion and Future Work

In this paper, I presented an interface which can assign arbitrary timbres to personal belongings, and enables the users to play music using them like musical instruments. In this interface, we can assign multiple timbres to each of the multiple targets simultaneously by real-time sound source separation and can play polyphonic music in spite of using one microphone only. Additionally, this system produces dynamically changing live sound as if the user is directly touching the actual object that produces the assigned timbre by modeling the physical sound generation process. Through the various experients, I verified the large potential of this interface as a musical instrument.

However, this interface leaves several problems to future work. Generally, hitting an object in different places produces different sounds. But this interface can't be considered this property until these sounds are registered individually in advance. Moreover, this interface can't assign different timbres to multiple sounds which are the same tone even if each sound originates from a separate position or different object. To address this problem is difficult using a microphone only without any other sensor. To improve the playability, considering the combination with other sensors is a future work.

In addition, I made a number of simplifications for simulating a physical sound. When a physical object produces a sound, its sound begins with two kinds of complex short-duration sounds, called "transient" and "acceleration noise," and continues into harmonic sound, called "tonal" afterwards [13]. My method is considered to be "tonal" only. However, "transient" and "acceleration noise" are important factors especially for the sound which has the particular timbre at the attack such as drums. To consider these effects, the introduction of several past methods such as the precomputed acceleration noise technique [14] and the synthesis technique using stationary wavelet transform and singular value decomposition [15] can be useful.

The non-linearlity is also an important factor. Many physical objects enter non-linear regimes when vibrating strongly, usually causing a spectral shift to higher frequencies, and generating a more complex sound overall. These complexities were not considered in this paper and are for future work.

There is also a problem in real-time sound source separation in that the NMF used in this paper assumes that the spectrum of a timbre is not time-varying. This assumption causes reducing the expressiveness of the separated sound. To address this, a non-parametric approach is considered of value [16]. However, the computational cost is too expensive to perform in real-time, so the construction of a speed-up algorithm is required.

References

[1] Sirdey, A., Derrien, O., Kronland-Martinet, R. and Aramaki, M.: Modal Analysis of Impact Sounds with ESPRIT in Gabor Transforms, *DAFx-11*, pp.387–392 (2011).

[2] van den Doel, K. and Pai, D.K.: Modal Synthesis for Vibrating Objects, *Audio Anecdotes III*, Greenbaum, (Ed.), Vol.10, No.2, pp.1–8, AK Peters, Natick, Massachusetts (2003).

[3] Fevotte, C. and Idier, J.: Algorithms for nonnegative matrix factorization with the beta-divergence, *Neural Computation, Article*, pp.1–24 (2011).

[4] MIDI, available from ⟨http://en.wikipedia.org/wiki/MIDI⟩.

[5] TableDrum, available from ⟨http://www.tabledrum.com/⟩.

[6] KORG WAVEDRUM, available from ⟨http://www.korg.co.jp/Product/Drum/WAVEDRUM/⟩.

[7] KORG WAVEDRUM Mini, available from ⟨http://www.korg.co.jp/Product/Drum/WAVEDRUMmini/⟩.

[8] Theremin, available from ⟨http://en.wikipedia.org/wiki/Theremin⟩.

[9] Effects unit, available from ⟨http://en.wikipedia.org/wiki/Effects_unit⟩.

[10] Merrill, D., Raffle, H. and Aimi, R.: The sound of touch: Physical manipulation of digital sound, *CHI'08: Proc. 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pp.739–472 (2008).

[11] Haykin, S.S. and Widrow, B.: Least-Mean-Square Adaptive Filters, *Wiley* (2003).

[12] Vincent, E., Gribonval, R. and Fevotte, C.: Performance measurement in blind audio source separation, *IEEE Trans. Audio, Speech and Language Processing* (2006).

[13] Stepanishen, P.: Transient Radiation, Crocker, M.J. (Ed.), *Handbook of Acoustics*, pp.119–126, NY: John Wiley and Sons Inc. (1998).

[14] Chadwick, J.N., Zheng, C. and James, D.L.: Precomputed Acceleration Noise for Improved Rigid-Body Sound, *ACM SIGGRAPH* (2012).

[15] Ahmad, W., Hacihabiboglu, H. and Kondoz, A.M.: Analysis-synthesis model for transient impact sounds by stationary wavelet transform and singular value decomposition, *International Computer Music Confer-*

ence (*ICMC'08*) (2008).
[16] Ozerov, A., Fevotte, C. and Charbit, M.: Factorial scaled hidden Markov model for polyphonic audio representation and source separation, *Proc. WASPAA* (2009).

Editor's Recommendation

At Interaction 2012, by the program committee of 87 members, the outstanding 18 papers among 43 submitted papers were adopted as general lecture presentation and 19 papers among 149 submitted papers were selected as finalists for interactive presentation. This paper, chosen from those 37 papers, gained a good assessment as a recommendable paper to journal by the program committee vote; therefore, as a journal editing chairperson, I certainly would like to recommend this paper.

(Interaction 2012 program chairperson, Homei Miyashita)

Kazuhiko Yamamoto was born in 1985. He received his Master of Design degree from Kyushu Univeresity in 2010 and has been engaged in the YAMAHA corporation since 2010. His research interests include computer graphics, numerical simulation, human computer interface, audio/image signal processing, and interactive art. He is a member of IPSJ, ASJ, IEEE, and ACM.