SPOTTING A QUERY PHRASE FROM POLYPHONIC MUSIC AUDIO SIGNALS BASED ON SEMI-SUPERVISED NONNEGATIVE MATRIX FACTORIZATION

Taro Masuda[1] **Kazuyoshi Yoshii**[2] **Masataka Goto**[3] **Shigeo Morishima**[1]
[1]Waseda University [2]Kyoto University
[3]National Institute of Advanced Industrial Science and Technology (AIST)
masutaro@suou.waseda.jp yoshii@i.kyoto-u.ac.jp m.goto@aist.go.jp shigeo@waseda.jp

ABSTRACT

This paper proposes a query-by-audio system that aims to detect temporal locations where a musical phrase given as a query is played in musical pieces. The "phrase" in this paper means a short audio excerpt that is not limited to a main melody (singing part) and is usually played by a single musical instrument. A main problem of this task is that the query is often buried in mixture signals consisting of various instruments. To solve this problem, we propose a method that can appropriately calculate the distance between a query and partial components of a musical piece. More specifically, gamma process nonnegative matrix factorization (GaP-NMF) is used for decomposing the spectrogram of the query into an appropriate number of basis spectra and their activation patterns. Semi-supervised GaP-NMF is then used for estimating activation patterns of the learned basis spectra in the musical piece by presuming the piece to partially consist of those spectra. This enables distance calculation based on activation patterns. The experimental results showed that our method outperformed conventional matching methods.

1. INTRODUCTION

Over a decade, a lot of effort has been devoted to developing music information retrieval (MIR) systems that aim to find musical pieces of interest by using audio signals as the query. For example, there are many similarity-based retrieval systems that can find musical pieces having similar acoustic features to those of the query [5,13,21,22]. Audio fingerprinting systems, on the other hand, try to find a musical piece that exactly matches the query by using acoustic features robust to audio-format conversion and noise contamination [6,12,27]. Query-by-humming (QBH) systems try to find a musical piece that includes the melody specified by users' singing or humming [19]. Note that in general information of musical scores [9,16,23,31,39] (such as MIDI files) or some speech corpus [36] should be prepared for a music database in advance of QBH. To overcome this limitation, some studies tried to automatically extract main melodies from music audio signals included in a database [25,34,35]. Other studies employ chroma vectors to characterize a query and targeted pieces without the need of symbolic representation or transcription [2].

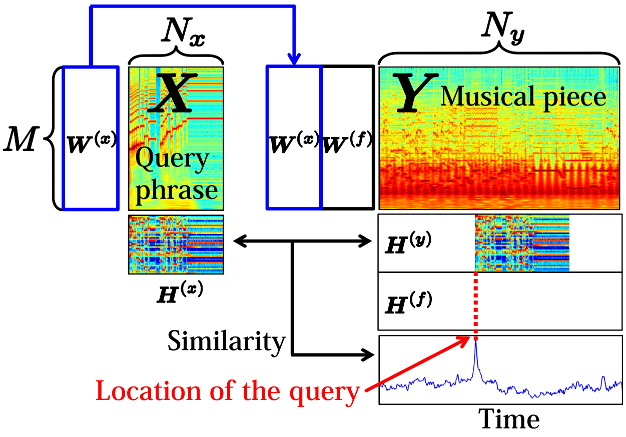

Figure 1. An overview of the proposed method.

We propose a task that aims to detect temporal locations at which phrases similar to the query phrase appear in different polyphonic musical pieces. The term "phrase" means a several-second musical performance (audio clip) usually played by a single musical instrument. Unlike QBH, our method needs no musical scores beforehand. A key feature of our method is that we aim to find short segments within musical pieces, not musical pieces themselves. There are several possible application scenarios in which both non-experts and music professionals enjoy the benefits of our system. For example, ordinary users could intuitively find a musical piece by playing just a characteristic phrase used in the piece even if the title of the piece is unknown or forgotten. In addition, composers could learn what kinds of arrangements are used in existing musical pieces that include a phrase specified as a query.

The major problem of our task lies in distance calculation between a query and short segments of a musical piece. One approach would be to calculate the symbolic distance between musical scores. However, this approach is impractical because even the state-of-the-art methods of automatic music transcription [4,11,17,29,38] work poorly for standard popular music. Conventional distance calculation based on acoustic features [5] is also inappropriate because acoustic features of a phrase are drastically distorted if other sounds are superimposed in a musical piece. In addition, since it would be more useful to find locations in which the same phrase is played by different instruments, we cannot heavily rely on acoustic features.

In this paper we propose a novel method that can perform phrase spotting by calculating the distance between a query and *partial* components of a musical piece. Our conjecture is that we could judge whether a phrase is included or not in a musical piece without perfect transcription, like the human ear can. More specifically, gamma process nonnegative matrix factorization (GaP-NMF) [14] is used for decomposing the spectrogram of a query into an appropriate number of basis spectra and their activation patterns. Semi-supervised GaP-NMF is then used for estimating activation patterns of the fixed basis spectra in a target musical piece by presuming the piece to *partially* consist of those spectra. This enables appropriate matching based on activation patterns of the basis spectra forming the query.

2. PHRASE SPOTTING METHOD

This section describes the proposed phrase-spotting method based on nonparametric Bayesian NMF.

2.1 Overview

Our goal is to detect the start times of a phrase in the polyphonic audio signal of a musical piece. An overview of the proposed method is shown in Figure 1. Let $\boldsymbol{X} \in \mathbb{R}^{M \times N_x}$ and $\boldsymbol{Y} \in \mathbb{R}^{M \times N_y}$ be the nonnegative power spectrogram of a query and that of a target musical piece, respectively. Our method consists of three steps. First, we perform NMF for decomposing the query $\boldsymbol{X}$ into a set of basis spectra $\boldsymbol{W}^{(x)}$ and a set of their corresponding activations $\boldsymbol{H}^{(x)}$. Second, in order to obtain temporal activations of $\boldsymbol{W}^{(x)}$ in the musical piece $\boldsymbol{Y}$, we perform another NMF whose basis spectra consist of a set of fixed basis spectra $\boldsymbol{W}^{(x)}$ and a set of unconstrained basis spectra $\boldsymbol{W}^{(f)}$ that are required for representing musical instrument sounds except for the phrase. Let $\boldsymbol{H}^{(y)}$ and $\boldsymbol{H}^{(f)}$ be sets of activations of $\boldsymbol{Y}$ corresponding to $\boldsymbol{W}^{(x)}$ and $\boldsymbol{W}^{(f)}$, respectively. Third, the similarity between the activation patterns $\boldsymbol{H}^{(x)}$ in the query and the activation patterns $\boldsymbol{H}^{(y)}$ in the musical piece is calculated. Finally, we detect locations of a phrase where the similarity takes large values.

There are two important reasons that "nonparametric" "Bayesian" NMF is needed. 1) It is better to automatically determine the optimal number of basis spectra according to the complexity of the query $\boldsymbol{X}$ and that of the musical piece $\boldsymbol{Y}$. 2) We need to put different prior distributions on $\boldsymbol{H}^{(y)}$ and $\boldsymbol{H}^{(f)}$ to put more emphasis on fixed basis spectra $\boldsymbol{W}^{(x)}$ than unconstrained basis spectra $\boldsymbol{W}^{(f)}$. If no priors are placed, the musical piece $\boldsymbol{Y}$ is often represented by using only unconstrained basis spectra $\boldsymbol{W}^{(f)}$. A key feature of our method is that we *presume* that the phrase is included in the musical piece when decomposing $\boldsymbol{Y}$. This means that we need to make use of $\boldsymbol{W}^{(x)}$ as much as possible for representing $\boldsymbol{Y}$. The Bayesian framework is a natural choice for reflecting such a prior belief.

2.2 NMF for Decomposing a Query

We use the gamma process NMF (GaP-NMF) [14] for approximating $\boldsymbol{X}$ as the product of a nonnegative vector $\boldsymbol{\theta} \in \mathbb{R}^{K_x}$ and two nonnegative matrices $\boldsymbol{W}^{(x)} \in \mathbb{R}^{M \times K_x}$ and $\boldsymbol{H}^{(x)} \in \mathbb{R}^{K_x \times N_x}$. More specifically, the original matrix $\boldsymbol{X}$ is factorized as follows:

$$X_{mn} \approx \sum_{k=1}^{K_x} \theta_k W_{mk}^{(x)} H_{kn}^{(x)}, \tag{1}$$

where θ_k is the overall gain of basis k, $W_{mk}^{(x)}$ is the power of basis k at frequency m, and $H_{kn}^{(x)}$ is the activation of basis k at time n. Each column of $\boldsymbol{W}^{(x)}$ represents a basis spectrum and each row of $\boldsymbol{H}^{(x)}$ represents an activation pattern of the basis over time.

2.3 Semi-supervised NMF for Decomposing a Musical Piece

We then perform semi-supervised NMF for decomposing the spectrogram of the musical piece $\boldsymbol{Y}$ by fixing a part of basis spectra with $\boldsymbol{W}^{(x)}$. The idea of giving $\boldsymbol{W}$ as a dictionary during inference has been widely adopted [3,7,15,18,24,26,28,30,33,38].

We formulate Bayesian NMF for representing the spectrogram of the musical piece $\boldsymbol{Y}$ by extensively using the fixed bases $\boldsymbol{W}^{(x)}$. To do this, we put different gamma priors on $\boldsymbol{H}^{(y)}$ and $\boldsymbol{H}^{(f)}$. The shape parameter of the gamma prior on $\boldsymbol{H}^{(y)}$ is much larger than that of the gamma prior on $\boldsymbol{H}^{(f)}$. Note that the expectation of the gamma distribution is proportional to its shape parameter.

2.4 Correlation Calculation between Activation Patterns

After the semi-supervised NMF is performed, we calculate the similarity between the activation patterns $\boldsymbol{H}^{(x)}$ in the query and the activation patterns $\boldsymbol{H}^{(y)}$ in a musical piece to find locations of the phrase. We expect that similar patterns appear in $\boldsymbol{H}^{(y)}$ when almost the same phrases are played in the musical piece even if those phrases are played by different instruments. More specifically, we calculate the sum of the correlation coefficients r at time n between $\boldsymbol{H}^{(x)}$ and $\boldsymbol{H}^{(y)}$ as follows:

$$r(n) = \frac{1}{K_x N_x} \sum_{k=1}^{K_x} \frac{\left(\boldsymbol{h}_{k1}^{(x)} - \overline{\boldsymbol{h}}_{k1}^{(x)}\right)^T \left(\boldsymbol{h}_{kn}^{(y)} - \overline{\boldsymbol{h}}_{kn}^{(y)}\right)}{\left\|\boldsymbol{h}_{k1}^{(x)} - \overline{\boldsymbol{h}}_{k1}^{(x)}\right\| \left\|\boldsymbol{h}_{kn}^{(y)} - \overline{\boldsymbol{h}}_{kn}^{(y)}\right\|}, \tag{2}$$

where

$$\boldsymbol{h}_{ki}^{(\cdot)} = \left[H_{ki}^{(\cdot)} \cdots H_{k(i+N_x-1)}^{(\cdot)}\right]^T, \tag{3}$$

$$\overline{\boldsymbol{h}}_{kn}^{(\cdot)} = \frac{1}{N_x} \sum_{j=1}^{N_x} H_{k(n+j-1)}^{(\cdot)} \times [1 \cdots 1]^T. \tag{4}$$

Finally, we detect a start frame n of the phrase by finding peaks of the correlation coefficients over time. This peak picking is performed based on the following thresholding process:

$$r(n) > \mu + 4\sigma, \tag{5}$$

where μ and σ denote the overall mean and standard deviation of $r(n)$, respectively, which were derived from all the musical pieces.

2.5 Variational Inference of GaP-NMF

This section briefly explains how to infer nonparametric Bayesian NMF [14], given a spectrogram $\boldsymbol{V} \in \mathbb{R}^{M \times N}$. We assume that $\boldsymbol{\theta} \in \mathbb{R}^{K}$, $\boldsymbol{W} \in \mathbb{R}^{M \times K}$, and $\boldsymbol{H} \in \mathbb{R}^{K \times N}$ are stochastically sampled according to a generative process. We choose a gamma distribution as a prior distribution on each parameter as follows:

$$\begin{aligned} p(W_{mk}) &= \mathrm{Gamma}\left(a^{(W)}, b^{(W)}\right), \\ p(H_{kn}) &= \mathrm{Gamma}\left(a^{(H)}, b^{(H)}\right), \\ p(\theta_k) &= \mathrm{Gamma}\left(\frac{\alpha}{K}, \alpha c\right), \end{aligned} \tag{6}$$

where α is a concentration parameter, K is a sufficiently large integer (ideally an infinite number) compared with the number of components in the mixed sound, and c is the inverse of the mean value of $\boldsymbol{V}$:

$$c = \left(\frac{1}{MN}\sum_m \sum_n V_{mn}\right)^{-1}. \tag{7}$$

We then use the generalized inverse-Gaussian (GIG) distribution as a posterior distribution as follows:

$$\begin{aligned} q(W_{mk}) &= \mathrm{GIG}\left(\gamma_{mk}^{(W)}, \rho_{mk}^{(W)}, \tau_{mk}^{(W)}\right), \\ q(H_{kn}) &= \mathrm{GIG}\left(\gamma_{kn}^{(H)}, \rho_{kn}^{(H)}, \tau_{kn}^{(H)}\right), \\ q(\theta_k) &= \mathrm{GIG}\left(\gamma_{k}^{(\theta)}, \rho_{k}^{(\theta)}, \tau_{k}^{(\theta)}\right). \end{aligned} \tag{8}$$

To estimate the parameters of these distributions, we first update other parameters, ϕ_{kmn}, ω_{mn}, using the following equations.

$$\phi_{kmn} = \mathbb{E}_q\left[\frac{1}{\theta_k W_{mk} H_{kn}}\right]^{-1}, \tag{9}$$

$$\omega_{mn} = \sum_k \mathbb{E}_q\left[\theta_k W_{mk} H_{kn}\right]. \tag{10}$$

After obtaining ϕ_{kmn} and ω_{mn}, we update the parameters of the GIG distributions as follows:

$$\begin{aligned} \gamma_{mk}^{(W)} = a^{(W)}, \quad \rho_{mk}^{(W)} &= b^{(W)} + \mathbb{E}_q[\theta_k]\sum_n \frac{\mathbb{E}_q[H_{kn}]}{\omega_{mn}}, \\ \tau_{mk}^{(W)} &= \mathbb{E}_q\left[\frac{1}{\theta_k}\right]\sum_n V_{mn}\phi_{kmn}^2 \mathbb{E}_q\left[\frac{1}{H_{kn}}\right], \end{aligned} \tag{11}$$

$$\begin{aligned} \gamma_{kn}^{(H)} = a^{(H)}, \quad \rho_{kn}^{(H)} &= b^{(H)} + \mathbb{E}_q[\theta_k]\sum_m \frac{\mathbb{E}_q[W_{mk}]}{\omega_{mn}}, \\ \tau_{kn}^{(H)} &= \mathbb{E}_q\left[\frac{1}{\theta_k}\right]\sum_m V_{mn}\phi_{kmn}^2 \mathbb{E}_q\left[\frac{1}{W_{mk}}\right], \end{aligned} \tag{12}$$

$$\begin{aligned} \gamma_{k}^{(\theta)} = \frac{\alpha}{K}, \quad \rho_{k}^{(\theta)} &= \alpha c + \sum_m \sum_n \frac{\mathbb{E}_q[W_{mk}H_{kn}]}{\omega_{mn}}, \\ \tau_{k}^{(\theta)} &= \sum_m \sum_n V_{mn}\phi_{kmn}^2 \mathbb{E}_q\left[\frac{1}{W_{mk}H_{kn}}\right]. \end{aligned} \tag{13}$$

The expectations of $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{\theta}$ are required in Eqs. (9) and (10). We randomly initialize the expectations of $\boldsymbol{W}$, $\boldsymbol{H}$, and $\boldsymbol{\theta}$ and iteratively update each parameter by using those formula. As the number of iterations increases, the value of $\mathbb{E}_q[\theta_k]$ over a certain level K^+ decreases. Therefore, if the value is 60 dB lower than $\sum_k \mathbb{E}_q[\theta_k]$, we remove the related parameters from consideration, which makes the calculation faster. Eventually, the number of effective bases, K^+, gradually reduces during iterations, suggesting that the appropriate number is automatically determined.

3. CONVENTIONAL MATCHING METHODS

We describe three kinds of conventional matching methods used for evaluation. The first and the second methods calculate the Euclidean distance between acoustic features (Section 3.1) and that between chroma vectors (Section 3.2), respectively. The third method calculates the Itakura-Saito (IS) divergence between spectrograms (Section 3.3).

3.1 MFCC Matching Based on Euclidean Distance

Temporal locations in which a phrase appears are detected by focusing on the acoustic distance between the query and a short segment extracted from a musical piece. In this study we use Mel-frequency cepstrum coefficients (MFCCs) as an acoustic feature, which have commonly been used in various research fields [1, 5]. More specifically, we calculate a 12-dimensional feature vector from each frame by using the Auditory Toolbox Version 2 [32]. The distance between two sequences of the feature vector extracted from the query and the short segment is obtained by accumulating the frame-wise Euclidean distance over the length of the query.

The above-mentioned distance is iteratively calculated by shifting the query frame by frame. Using a simple peak-picking method, we detect locations of the phrase in which the obtained distance is lower than $m - s$, where m and s denote the mean and standard deviation of the distance over all frames, respectively.

3.2 Chromagram Matching Based on Euclidean Distance

In this section, temporal locations in which a phrase appears are detected in the same manner as explained in Section 3.1. A difference is that we extracted a 12-dimentional chroma vector from each frame by using the MIRtoolbox [20]. In addition, we empirically defined the threshold of the peak-picking method as $m - 3s$.

3.3 DP Matching Based on Itakura-Saito Divergence

In this section, temporal locations in which a phrase appears are detected by directly calculating the Itakura-Saito (IS) divergence [8, 37] between the query $\boldsymbol{X}$ and the musical piece $\boldsymbol{Y}$. The use of the IS divergence is theoretically justified because the IS divergence poses a smaller penalty than standard distance measures such as the Euclidean distance and the Kullback-Leibler (KL) divergence when the power spectrogram of the query is included in that of the musical piece.

To efficiently find phrase locations, we use a dynamic programming (DP) matching method based on the IS divergence. First, we make a distance matrix $\boldsymbol{D} \in \mathbb{R}^{N_x \times N_y}$ in which each cell $D(i,j)$ is the IS divergence between the i-th frame of $\boldsymbol{X}$ and the j-th frame of $\boldsymbol{Y}$ ($1 \le i \le N_x$ and $1 \le j \le N_y$). $D(i,j)$ is given by

$$D(i,j) = \mathcal{D}_{\mathrm{IS}}(\boldsymbol{X}_i | \boldsymbol{Y}_j) = \sum_m \left(-\log \frac{X_{mi}}{Y_{mj}} + \frac{X_{mi}}{Y_{mj}} - 1 \right), \tag{14}$$

where m indicates a frequency-bin index. We then let $\boldsymbol{E} \in \mathbb{R}^{N_x \times N_y}$ be a cumulative distance matrix. First, $\boldsymbol{E}$ is initialized as $E(1,j) = 0$ for any j and $E(i,1) = \infty$ for any i. $E(i,j)$ can be sequentially calculated as follows:

$$E(i,j) = \min \left\{ \begin{array}{l} 1)\ E(i-1,j-2) + 2D(i,j-1) \\ 2)\ E(i-1,j-1) + D(i,j) \\ 3)\ E(i-2,j-1) + 2D(i-1,j) \end{array} \right\} + D(i,j). \tag{15}$$

Finally, we can obtain $E(N_x, j)$ that represents the distance between the query and a phrase ending at the j-th frame in the musical piece. We let $\boldsymbol{C} \in \mathbb{R}^{N_x \times N_y}$ be a cumulative cost matrix. According to the three cases 1), 2), and 3), $C(i,j)$ is obtained as follows:

$$C(i,j) = \left\{ \begin{array}{l} 1)\ C(i-1,j-2) + 3 \\ 2)\ C(i-1,j-1) + 2 \\ 3)\ C(i-2,j-1) + 3. \end{array} \right. \tag{16}$$

This means that the length of a phrase is allowed to range from one half to two times of the query length.

Phrase locations are determined by finding the local minima of the regularized distance given by $\frac{E(N_x,j)}{C(N_x,j)}$. More specifically, we detect locations in which values of the obtained distance are lower than $M - S/10$, where M and S denote the median and standard deviation of the distance over all frames, respectively. A reason that we use the median for thresholding is that the distance sometimes takes an extremely large value (outlier). The mean of the distance tends to be excessively biased by such an outlier. In addition, we ignore values of the distance which are more than 10^6 when calculating S for practical reasons (almost all values of $\frac{E(N_x,j)}{C(N_x,j)}$ range from 10^3 to 10^4). Once the end point is detected, we can also obtain the start point of the phrase by simply tracing back along the path from the end point.

4. EXPERIMENTS

This section reports comparative experiments that were conducted for evaluating the phrase-spotting performances of the proposed method described in Section 2 and the three conventional methods described in Section 3.

4.1 Experimental Conditions

The proposed method and the three conventional methods were tested under three different conditions: 1) Exactly the same phrase specified as a query was included in a musical piece (exact match). 2) A query was played by a different kind of musical instruments (timbre change). 3) A query was played in a faster tempo (tempo change).

We chose four musical pieces (RWC-MDB-P-2001 No.1, 19, 42, and 77) from the RWC Music Database: Popular Music [10]. We then prepared 50 queries: 1) 10 were short segments excerpted from original multi-track recordings of the four pieces. 2) 30 queries were played by three kinds of musical instruments (nylon guitar, classic piano, and strings) that were different from those originally used in the four pieces. 3) The remaining 10 queries were played by the same instruments as original ones, but their tempi were 20% faster. Each query was a short performance played by a single instrument and had a duration ranging from 4 s to 9 s. Note that those phrases were not necessarily salient (not limited to main melodies) in musical pieces. We dealt with monaural audio signals sampled at 16 kHz and applied the wavelet transform by shifting short-time frames with an interval of 10 ms. The reason that we did not use short-time Fourier transform (STFT) was to attain a high resolution in a low frequency band. We determined the standard deviation of a Gabor wavelet function to 3.75 ms (60 samples). The frequency interval was 10 cents and the frequency ranged from 27.5 (A1) to 8000 (much higher than C8) Hz.

When a query was decomposed by NMF, the hyperparameters were set as $\alpha = 1$, $K = 100$, $a^{(W)} = b^{(W)} = a^{(H)} = 0.1$, and $b^{(H^{(x)})} = c$. When a musical piece was decomposed by semi-supervised NMF, the hyperparameters were set as $a^{(W)} = b^{(W)} = 0.1$, $a^{(H^{(y)})} = 10$, $a^{(H^{(f)})} = 0.01$, and $b^{(H)} = c$. The inverse-scale parameter $b^{(H)}$ was adjusted to the empirical scale of the spectrogram of a target audio signal. Also note that using smaller values of $a^{(\cdot)}$ makes parameters sparser in an infinite space.

To evaluate the performance of each method, we calculated the average F-measure, which has widely been used in the field of information retrieval. The precision rate was defined as a proportion of the number of correctly-found

	Precision (%)	Recall (%)	F-measure (%)
MFCC	24.8	35.0	29.0
Chroma	33.4	61.0	43.1
DP	1.9	55.0	3.6
Proposed	53.6	63.0	57.9

Table 1. Experimental results in a case that exactly the same phrase specified as a query was included in a musical piece.

	Precision (%)	Recall (%)	F-measure (%)
MFCC	0	0	0
Chroma	18.1	31.7	23.0
DP	1.1	66.3	6.2
Proposed	26.9	56.7	36.5

Table 2. Experimental results in a case that a query was played by a different kind of instruments.

	Precision (%)	Recall (%)	F-measure (%)
MFCC	0	0	0
Chroma	12.0	19.0	14.7
DP	0.5	20.0	2.7
Proposed	15.8	45.0	23.4

Table 3. Experimental results in a case that the query phrases was played in a faster tempo.

phrases to that of all the retrieved phrases. The recall rate was defined as a proportion of the number of correctly-found phrases to that of all phrases included in the database (each query phrase was included only in one piece of music). Subsequently, we calculated the F-measure F by $F = \frac{2PR}{P+R}$, where P and R denote the precision and recall rates, respectively. We regarded a detected point as a correct one when its error is within 50 frames (500 ms).

4.2 Experimental Results

Tables 1–3 show the accuracies obtained by the four methods under each condition. We confirmed that our method performed much better than the conventional methods in terms of accuracy. Figure 2 shows the value of $r(n)$ obtained from a musical piece in which a query phrase (originally played by the saxophone) is included. We found that the points at which the query phrase starts were correctly spotted by using our method. Although the MFCC-based method could retrieve some of the query phrases in the exact-match condition, it was not robust to timbre change and tempo change. The DP matching method, on the other hand, could retrieve very few correct points because the IS divergence was more sensitive to volume change than the similarity based on spectrograms. Although local minima of the cost function often existed at correct points, those minima were not sufficiently clear because it was difficult to detect the end point of the query from the spectrogram of a mixture audio signal. The chroma-based method worked better than the other conventional methods. However, it did not outperform the proposed method since the chroma-based method often detected false locations including a similar chord progression.

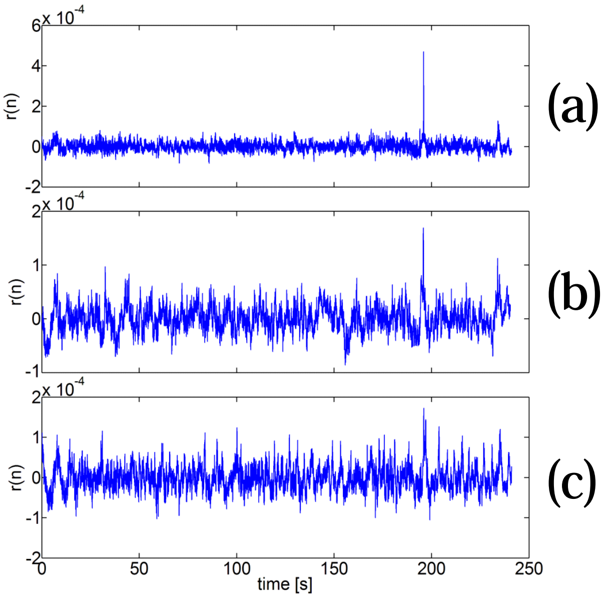

Figure 2. Sum of the correlation coefficients $r(n)$. The target piece was RWC-MDB-P-2001 No.42. (a) The query was exactly the same as the target saxophone phrase. (b) The query was played by strings. (c) The query was played 20% faster than the target.

Although our method worked best of the four, the accuracy of the proposed method should be improved for practical use. A major problem is that the precision rate was relatively lower than the recall rate. Wrong locations were detected when queries were played in *staccato* manner because many false peaks appeared at the onset of *staccato* notes.

As for computational cost, it took 29746 seconds to complete the retrieval of a single query by using our method. This was implemented in C++ on a 2.93 GHz Intel Xeon Windows 7 with 12 GB RAM.

5. CONCLUSION AND FUTURE WORK

This paper presented a novel query-by-audio method that can detect temporal locations where a phrase given as a query appears in musical pieces. Instead of pursuing perfect transcription of music audio signals, our method used nonnegative matrix factorization (NMF) for calculating the distance between the query and partial components of each musical piece. The experimental results showed that our method performed better than conventional matching methods. We found that our method has a potential to find correct locations in which a query phrase is played by different instruments (timbre change) or in a faster tempo (tempo change).

Future work includes improvement of our method, especially under the timbre-change and tempo-change conditions. One promising solution would be to classify basis spectra of a query into instrument-dependent bases (*e.g.*, noise from the guitar) and common ones (*e.g.*, harmonic spectra corresponding to musical notes) or to create an universal set of basis spectra. In addition, we plan to reduce the computational cost of our method based on nonparametric Bayesian NMF.

Acknowledgment: This study was supported in part by the JST OngaCREST project.

6. REFERENCES

[1] J. J. Aucouturier and F. Pachet. Music Similarity Measures: What's the Use?, *ISMIR*, pp. 157–163, 2002.

[2] C. de la Bandera, A. M. Barbancho, L. J. Tardón, S. Sammartino, and I. Barbancho. Humming Method for Content-Based Music Information Retrieval, *ISMIR*, pp. 49–54, 2011.

[3] L. Benaroya, F. Bimbot, and R. Gribonval. Audio Source Separation with a Single Sensor, *IEEE Trans. on ASLP*, 14(1):191–199, 2006.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic Music Transcription: Breaking the Glass Ceiling, *ISMIR*, pp. 379–384, 2012.

[5] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures, *Computer Music Journal*, 28(2):63–76, 2004.

[6] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A Review of Audio Fingerprinting, *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 41(3):271–284, 2005.

[7] Z. Duan, G. J. Mysore, and P. Smaragdis. Online PLCA for Real-Time Semi-supervised Source Separation, *Latent Variable Analysis and Signal Separation*, Springer Berlin Heidelberg, pp. 34–41, 2012.

[8] A. El-Jaroudi and J. Makhoul. Discrete All-Pole Modeling, *IEEE Trans. on Signal Processing*, 39(2):411–423, 1991.

[9] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by Humming: Musical Information Retrieval in an Audio Database, *ACM Multimedia*, pp. 231–236, 1995.

[10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases, *ISMIR*, pp. 287–288, 2002.

[11] G. Grindlay and D. P. W. Ellis. A Probabilistic Subspace Model for Multi-instrument Polyphonic Transcription, *ISMIR*, pp. 21–26, 2010.

[12] J. Haitsma and T. Kalker. A Highly Robust Audio Fingerprinting System, *ISMIR*, pp. 107–115, 2002.

[13] M. Helén and T. Virtanen. Audio Query by Example Using Similarity Measures between Probability Density Functions of Features, *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.

[14] M. D. Hoffman, D. M. Blei, and P. R. Cook. Bayesian Nonparametric Matrix Factorization for Recorded Music, *ICML*, pp. 439–446, 2010.

[15] X. Jaureguiberry, P. Leveau, S. Maller, and J. J. Burred. Adaptation of source-specific dictionaries in Non-Negative Matrix Factorization for source separation, *ICASSP*, pp. 5–8, 2011.

[16] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody Retrieval with Humming, *ICMC*, pp.349–351, 1993.

[17] H. Kameoka, K. Ochiai, M. Nakano, M. Tsuchiya, and S. Sagayama. Context-free 2D Tree Structure Model of Musical Notes for Bayesian Modeling of Polyphonic Spectrograms, *ISMIR*, pp. 307–312, 2012.

[18] H. Kirchhoff, S. Dixon, and A. Klapuri. Multi-Template Shift-variant Non-negative Matrix Deconvolution for Semi-automatic Music Transcription, *ISMIR*, pp. 415–420, 2012.

[19] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, and V. Athitsos. A Survey of Query-By-Humming Similarity Methods, *International Conference on PETRA*, 2012.

[20] O. Lartillot and P. Toiviainen. A Matlab Toolbox for Musical Feature Extraction from Audio, *DAFx*, pp. 237–244, 2007.

[21] T. Li and M. Ogihara. Content-based Music Similarity Search and Emotion Detection, *ICASSP*, Vol. 5, pp. 705–708, 2004.

[22] B. Logan and A. Salomon. A Music Similarity Function Based on Signal Analysis, *International Conference on Multimedia and Expo (ICME)*, pp. 745–748, 2001.

[23] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the Digital Music Library: Tune Retrieval from Acoustic Input, *ACM international conference on Digital libraries*, pp. 11–18, 1996.

[24] G. J. Mysore and P. Smaragdis. A Non-negative Approach to Semi-supervised Separation of Speech from Noise with the Use of Temporal Dynamics, *ICASSP*, pp. 17–20, 2011.

[25] T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka. Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming, *ISMIR*, pp. 211–218, 2001.

[26] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot. One Microphone Singing Voice Separation Using Source-adapted Models, *WASPAA*, pp. 90–93, 2005.

[27] M. Ramona and G. Peeters. AudioPrint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme, *ICASSP*, pp. 818–822, 2013.

[28] S. T. Roweis. One Microphone Source Separation, *Advances in Neural Information Processing Systems*, Vol. 13, MIT Press, pp. 793–799, 2001.

[29] M. Ryynänen and A. Klapuri. Automatic Bass Line Transcription from Streaming Polyphonic Audio, *ICASSP*, pp. IV–1437–1440, 2007.

[30] M. N. Schmidt and R. K. Olsson. Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization, *Interspeech*, pp. 1652–1655, 2006.

[31] J. Shifrin, B. Pardo, C. Meek, and W. Birmingham. HMM-Based Musical Query Retrieval, *ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 295–300, 2002.

[32] M. Slaney. Auditory Toolbox Version 2, *Technical Report #1998-010*, Interval Research Corporation, 1998.

[33] P. Smaragdis, B. Raj, and M. Shashanka. Supervised and Semi-supervised Separation of Sounds from Single-Channel Mixtures, *Independent Component Analysis and Signal Separation*, Springer Berlin Heidelberg, pp. 414–421, 2007.

[34] C. J. Song, H. Park, C. M. Yang, S. J. Jang, and S. P. Lee. Implementation of a Practical Query-by-Singing/Humming (QbSH) System and Its Commercial Applications, *IEEE Trans. on Consumer Electronics*, 59(2):407–414, 2013.

[35] J. Song, S. Y. Bae, and K. Yoon. Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System, *ISMIR*, pp. 133–139, 2002.

[36] C. C. Wang, J. S. R. Jang, and W. Wang. An Improved Query by Singing/Humming System Using Melody and Lyrics Information, *ISMIR*, pp. 45–50, 2010.

[37] B. Wei and J. D. Gibson. Comparison of Distance Measures in Discrete Spectral Modeling, *IEEE DSP Workshop*, 2000.

[38] F. Weninger, C. Kirst, B. Schuller, and H. J. Bungartz. A Discriminative Approach to Polyphonic Piano Note Transcription Using Supervised Non-negative Matrix Factorization, *ICASSP*, pp. 26–31, 2013.

[39] Y. Zhu and D. Shasha. Query by Humming: a Time Series Database Approach, *SIGMOD*, 2003.