REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation

Zafar Rafii, *Student Member, IEEE,* and Bryan Pardo, *Member, IEEE*

***Abstract*—Repetition is a core principle in music. Many musical pieces are characterized by an underlying repeating structure over which varying elements are superimposed. This is especially true for pop songs where a singer often overlays varying vocals on a repeating accompaniment. On this basis, we present the *REpeating Pattern Extraction Technique (REPET)*, a novel and simple approach for separating the repeating "background" from the non-repeating "foreground" in a mixture. The basic idea is to identify the periodically repeating segments in the audio, compare them to a repeating segment model derived from them, and extract the repeating patterns via time-frequency masking. Experiments on data sets of 1,000 song clips and 14 full-track real-world songs showed that this method can be successfully applied for music/voice separation, competing with two recent state-of-the-art approaches. Further experiments showed that REPET can also be used as a preprocessor to pitch detection algorithms to improve melody extraction.**

***Index Terms*—Melody extraction, music structure analysis, music/voice separation, repeating patterns.**

I. Introduction

REPETITION "is the basis of music as an art" [1]. Music theorists such as Schenker had shown that the concept of *repetition* is very important for the analysis of structure in music. In Music Information Retrieval (MIR), researchers used repetition/similarity mainly for audio segmentation and summarization, and sometimes for rhythm estimation (see Section I-A). In this work, we show that we can also use the analysis of the repeating structure in music for source separation.

The ability to efficiently separate a song into its music and voice components would be of great interest for a wide range of applications, among others instrument/vocalist identification, pitch/melody extraction, audio post processing, and karaoke gaming. Existing methods in music/voice separation do not explicitly use the analysis of the repeating structure as a basis for separation (see Section I-B). We take a fundamentally different approach to separating the lead melody from the background accompaniment: find the repeating patterns in the audio and extract them from the non-repeating elements.

Manuscript received December 07, 2011; revised June 15, 2012; accepted August 02, 2012. Date of publication August 15, 2012; date of current version October 18, 2012. This work was supported by the National Science Foundation (NSF) under Grant IIS-0812314. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jingdong Chen.

The authors are with the Department of Electrical Engineering and Computer Science, Ford Motor Company Engineering Design Center, Northwestern University, Evanston, IL 60208 USA (e-mail: zafarrafii@u.northwestern.edu; pardo@northwestern.edu).

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

Digital Object Identifier 10.1109/TASL.2012.2213249

The justification for this approach is that many musical pieces are composed of structures where a singer overlays varying lyrics on a repeating accompaniment. Examples include singing different verses over the same chord progression or rapping over a repeated drum loop. The idea is to identify the periodically repeating patterns in the audio (e.g., a guitar riff or a drum loop), and then separate the repeating "background" from the non-repeating "foreground" (typically the vocal line). This is embodied in an algorithm called *REpeating Pattern Extraction Technique (REPET)* (see Section I-C).

In Section II, we outline the REPET algorithm. In Section III, we evaluate REPET on a data set of 1,000 song clips against a recent competitive method. In Section IV, we evaluate REPET on the same data set against another recent competitive method; we also investigate potential improvements to REPET and analyze the interactions between length, repetitions, and performance in REPET. In Section V, we propose a simple procedure to extend REPET to longer musical pieces, and evaluate it on a new data set of 14 full-track real-world songs. In Section VI, we evaluate REPET as a preprocessor for two pitch detection algorithms to improve melody extraction. In Section VII, we conclude this article.

A. Music Structure Analysis

In music theory, Schenker asserted that repetition is what gives rise to the concept of the motive, which is defined as the smallest structural element within a musical piece [1]. Ruwet used repetition as a criterion for dividing music into small parts, revealing the syntax of the musical piece [2]. Ockelford argued that repetition/imitation is what brings order to music, and order is what makes music aesthetically pleasing [3].

More recently, researchers in MIR have recognized the importance of repetition/similarity for music structure analysis. For visualizing the musical structure, Foote introduced the *similarity matrix*, a two-dimensional matrix where each bin measures the (dis)similarity between any two instances of the audio [4]. The similarity matrix (or its dual, the distance matrix) can be built from different features, such as the Mel-Frequency Cepstrum Coefficients (MFCC) [4]–[7], the spectrogram [8], [9], the chromagram [7], [10]–[12], the pitch contour [11], [13], or other features [7], [11], [12], as long as similar sounds yield similarity in the feature space. Different similarity (or distance) functions can also be used, such as the dot product [4], [10], the cosine similarity [5], [8], [9], the Euclidean distance [6], [12], or other functions [11], [13].

Foote suggested to use the similarity matrix for tasks such as audio segmentation [8], music summarization [5], and beat estimation [9]. For example, he generated a novelty curve by

identifying changes in local self-similarity in a similarity matrix built from the spectrogram [8]. Other audio segmentation methods include Jensen who used similarity matrices built from features related to rhythm, timbre, and harmony [12].

Bartsch detected choruses in popular music by analyzing the structural redundancy in a similarity matrix built from the chromagram [10]. Other audio thumbnailing methods include Cooper *et al.* who built a similarity matrix using MFCCs [5].

Dannenberg *et al.* generated a description of the musical structure related to the AABA form by using similarity matrices built from a monophonic pitch estimation [13], and also the chromagram and a polyphonic transcription [11]. Other music summarization methods include Peeters who built similarity matrices using MFCCs, the chromagram, and dynamic rhythmic features [7].

Foote *et al.* developed the *beat spectrum*, a measure of acoustic self-similarity as a function of the time lag, by using a similarity matrix built from the spectrogram [9]. Other beat estimation methods include Pikrakis *et al.* who built a similarity matrix using MFCCs [6].

For a thorough review on music structure analysis, the reader is referred to [7], [14] and [15].

B. Music/Voice Separation

Music/voice separation methods typically first identify the vocal/non-vocal segments, and then use a variety of techniques to separate the lead vocals from the background accompaniment, including spectrogram factorization, accompaniment model learning, and pitch-based inference techniques.

Vembu *et al.* first identified the vocal and non-vocal regions by computing features such as MFCCs, Perceptual Linear Predictive coefficients (PLP), and Log Frequency Power Coefficients (LFPC), and using classifiers such as Neural Networks (NN) and Support Vector Machines (SVM). They then used Non-negative Matrix Factorization (NMF) to separate the spectrogram into vocal and non-vocal basic components [16]. However, for an effective separation, NMF requires a proper initialization and the right number of components.

Raj *et al.* used a priori known non-vocal segments to train an accompaniment model based on a Probabilistic Latent Component Analysis (PLCA). They then fixed the accompaniment model to learn the vocal parts [17]. Ozerov *et al.* first performed a vocal/non-vocal segmentation using MFCCs and Gaussian Mixture Models (GMM). They then trained Bayesian models to adapt an accompaniment model learned from the non-vocal segments [18]. However, for an effective separation, such accompaniment model learning techniques require a sufficient amount of non-vocal segments and an accurate vocal/non-vocal prior segmentation.

Li *et al.* performed a vocal/non-vocal segmentation using MFCCs and GMMs. They then used a predominant pitch estimator on the vocal segments to extract the pitch contour, which was finally used to separate the vocals via binary masking [19]. Ryynänen *et al.* proposed to use a melody transcription method to estimate the MIDI notes and the fundamental frequency trajectory of the vocals. They then used sinusoidal models to estimate and remove the vocals from the accompaniment [20]. However, such pitch-based inference techniques cannot deal with unvoiced vocals and furthermore, the harmonic structure of the instruments may interfere.

Virtanen *et al.* proposed a hybrid method where they first used a pitch-based inference technique, followed by a binary masking to extract the harmonic structure of the vocals. They then used NMF on the remaining spectrogram to learn an accompaniment model [21].

Hsu *et al.* first used a Hidden Markov Model (HMM) to identify accompaniment, voiced, and unvoiced segments. They then used the pitch-based inference method of Li *et al.* to separate the voiced vocals [19], while the pitch contour was derived from the predominant pitch estimation algorithm of Dressler [22]. In addition, they proposed a method to separate the unvoiced vocals based on GMMs and a method to enhance the voiced vocals based on spectral subtraction [23]. This is a state-of-the-art system we compare to in our evaluation.

Durrieu *et al.* proposed to model a mixture as the sum of a signal of interest (lead) and a residual (background), where the background is parameterized as an unconstrained NMF model, and the lead as a source/filter model. They then separated the lead from the background by estimating the parameters of their model in an iterative way using an NMF-based framework. In addition, they incorporated a white noise spectrum in their decomposition to capture the unvoiced components [24]. This is a state-of the art system we compare to in our evaluation.

C. Proposed Method

We present the *REpeating Pattern Extraction Technique (REPET)*, a simple and novel approach for separating a repeating background from a non-repeating foreground. The basic idea is to identify the periodically repeating segments, compare them to a repeating segment model, and extract the repeating patterns via time-frequency masking (see Section II).

The justification for this approach is that many musical pieces can be understood as a repeating background over which a lead is superimposed that does not exhibit any immediate repeating structure. For excerpts with a relatively stable repeating background (e.g., 10 second verse), we show that REPET can be successfully applied for music/voice separation (see Sections III and IV). For full-track songs, the repeating background is likely to show variations over time (e.g., verse followed by chorus). We therefore also propose a simple procedure to extend the method to longer musical pieces, by applying REPET on local windows of the signal over time (see Section V).

Unlike other separation approaches, REPET does not depend on particular statistics (e.g., MFCC or chroma features), does not rely on complex frameworks (e.g., pitch-based inference techniques or source/filter modeling), and does not require preprocessing (e.g., vocal/non-vocal segmentation or prior training). Because it is only based on self-similarity, it has the advantage of being simple, fast, and blind. It is therefore, completely and easily automatable.

A parallel can be drawn between REPET and background subtraction. Background subtraction is the process of separating a background scene from foreground objects in a sequence of video frames. The basic idea is the same, but the approaches are different. In background subtraction, no period estimation nor temporal segmentation are needed since the video frames

already form a periodic sample. Also, the variations of the background have to be handled in a different manner since they involve characteristics typical of images. For a review on background subtraction, the reader is referred to [25].

REPET bears some similarity with the drum sound recognizer of Yoshii *et al.* [26]. Their method iteratively updates time-frequency templates corresponding to drum patterns in the spectrogram, by taking the element-wise median of the patterns that are similar to a template, until convergence. As a comparison, REPET directly derives a whole repeating segment model by taking the element-wise median of all the periodically repeating segments in the spectrogram (see Section II).

Although REPET was defined here as a method for separating the repeating background from the non-repeating foreground in a musical mixture, it could be generalized to any kind of repeating patterns. In particular, it could be used in Active Noise Control (ANC) for removing periodic interferences. Applications include canceling periodic interferences in electrocardiography (e.g., the power-line interference), or in speech signals (e.g., a pilot communicating by radio from an aircraft) [27]. While REPET can be applied for periodic interferences removal, ANC algorithms cannot be applied for music/voice separation due to the simplicity of the models used. For a review on ANC, the reader is referred to [27].

The idea behind REPET that repetition can be used for source separation has also been supported by recent findings in psychoacoustics. McDermott *et al.* established that the human auditory system is able to segregate individual sources by identifying them as repeating patterns embedded in the acoustic input, without requiring prior knowledge of the source properties [28]. Through a series of hearing studies, they showed that human listeners are able to identify a never-heard-before target sound if it repeats within different mixtures.

II. REPET

In this section, we detail the *REpeating Pattern Extraction Technique (REPET)*. The method can be summarized in three stages: identification of the repeating period (Section II-A), modeling of the repeating segment (Section II-B), and extraction of the repeating patterns (Section II-C). Compared to the original REPET introduced in [29], we propose an enhanced repeating period estimation algorithm, an improved repeating segment modeling, and an alternate way for building the time-frequency masking. In addition, we also propose a simple procedure to extend the method to longer musical pieces (see Section V-B).

A. Repeating Period Identification

Periodicities in a signal can be found by using the *autocorrelation*, which measures the similarity between a segment and a lagged version of itself over successive time intervals.

Given a mixture signal x, we first calculate its Short-Time Fourier Transform (STFT) X, using half-overlapping Hamming windows of N samples. We then derive the magnitude spectrogram V by taking the absolute value of the elements of X, after discarding the symmetric part, while keeping the DC component. We then compute the autocorrelation of each row of the power spectrogram V^2 (element-wise square of V) and obtain

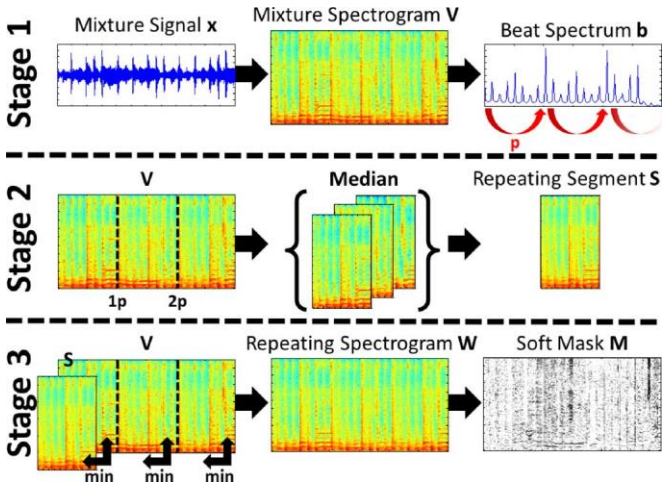

Fig. 1. Overview of the REPET algorithm. Stage 1: calculation of the beat spectrum and estimation of the repeating period. Stage 2: segmentation of the mixture spectrogram and computation of the repeating segment model. Stage 3: derivation of the repeating spectrogram model and building of the soft time-frequency mask.

the matrix B. We use V^2 to emphasize the appearance of peaks of periodicity in B. If the mixture signal x is stereo, V^2 is averaged over the channels. The overall acoustic self-similarity b of x is obtained by taking the mean over the rows of B. We finally normalize b by its first term (lag 0). The calculation of b is shown in (1).

$$B(i,j) = \frac{1}{m-j+1} \sum_{k=1}^{m-j+1} V(i,k)^2 \, V(i,k+j-1)^2$$

$$b(j) = \frac{1}{n} \sum_{i=1}^{n} B(i,j) \qquad \text{then } b(j) = \frac{b(j)}{b(1)}$$

$$\text{for } i = 1 \ldots n \,(\text{frequency}) \text{ where } n = \frac{N}{2} + 1$$

$$\text{for } j = 1 \ldots m \,(\text{lag}) \text{ where } m = \#\text{time frames}. \qquad (1)$$

The idea is similar to the *beat spectrum* introduced in [9], except that no similarity matrix is explicitly calculated here and the dot product is used in lieu of the cosine similarity. Pilot experiments showed that this method allows for a clearer visualization of the beat structure in x. For simplicity, we will refer to b as the beat spectrum for the remainder of the paper.

Once the beat spectrum is calculated, the first term which measures the similarity of the whole signal with itself (lag 0) is discarded. If repeating patterns are present in x, b would form peaks that are periodically repeating at different levels, revealing the underlying hierarchical repeating structure of the mixture, as exemplified in the top row of Fig. 1.

We use a simple procedure for automatically estimating the repeating period p. The basic idea is to find which period in the beat spectrum has the highest mean accumulated energy over its integer multiples. For each possible period j in b, we check if its integer multiples i (i.e., j, $2j$, $3j$, etc.) correspond to the highest peaks in their respective neighborhoods $[i - \Delta, i + \Delta]$, where Δ is a variable distance parameter, function of j. If they do, we sum their values, minus the mean of the given neighborhood to filter any possible "noisy background."

Algorithm 1 Find repeating period p from beat spectrum b

$l \leftarrow$ length of b after discarding the longest $1/4$ of lags
$\delta \leftarrow$ fixed deviation for possible shifted peaks
$J \leftarrow$ empty array of length $\lfloor l/3 \rfloor$
for each possible period j in the first 1/3 of b **do**
$\Delta \leftarrow \lfloor 3j/4 \rfloor$, $I \leftarrow 0$
for each possible integer multiple i of j in b **do**
$h \leftarrow \underset{k \in [i-\delta, i+\delta]}{\operatorname{argmax}} b(k)$
if $h = \underset{k \in [i-\Delta, i+\Delta]}{\operatorname{argmax}} b(k)$ **then**
$I \leftarrow I + b(h)$
end if
end for
$J(j) \leftarrow I / \lfloor l/j \rfloor$
end for
$p \leftarrow \arg\max_j J(j)$

We then divide this sum by the total number of integer multiples of j found in b, leading to a mean energy value for each period j. We define the repeating period p as the period j that gives the largest mean value. This helps to find the period of the strongest repeating peaks in b, corresponding to the period of the underlying repeating structure in x, while avoiding lower-order (periods of smaller repeating patterns) and higher-order errors (multiples of the repeating period).

The longest lag terms of the autocorrelation are often unreliable, since the further we get in time, the fewer coefficients are used to compute the similarity. Therefore, we choose to ignore the values in the longest 1/4 of lags in b. Because we want to have at least three segments to build the repeating segment model, we limit our choice of periods to those periods that allow three full cycles in the remaining portion of b.

We set the distance parameter Δ to $\lfloor 3j/4 \rfloor$ for each possible period j, where $\lfloor . \rfloor$ represents the floor function. This creates a window around a peak that is wide, but not so wide that it includes other peaks at multiples of j. Because of tempo deviations, the repeating peaks in b might not be exact integer multiples of j, so we also introduce a fixed deviation parameter δ that we set to 2 lags. This means that when looking for the highest peak in the neighborhood $[i-\Delta, i+\Delta]$, we assume that the value of the corresponding integer multiple i is the maximum of the local interval $[i - \delta, i + \delta]$. The estimation of the repeating period p is described in Algorithm 1. The calculation of the beat spectrum b and the estimation of the repeating period are illustrated in the top row of Fig. 1.

B. Repeating Segment Modeling

Once the repeating period p is estimated from the beat spectrum b, we use it to evenly time-segment the spectrogram V into r segments of length p. We define the repeating segment model S as the element-wise median of the r segments, as exemplified in the middle row of Fig. 1. The calculation of the repeating segment model S is shown in (2).

$$S(i,l) = \underset{k=1\ldots r}{median}\{V(i, l+(k-1)p)\} \quad \text{for } i = 1\ldots n \text{ (frequency) and } l = 1\ldots p \text{ (time)} \quad \text{where } p = \text{period length and } r = \#\text{ segments.} \tag{2}$$

The rationale is that, assuming that the non-repeating foreground ($\approx$ voice) has a sparse and varied time-frequency representation compared with the time-frequency representation of the repeating background ($\approx$ music) – a reasonable assumption for voice in music, time-frequency bins with little deviation at period p would constitute a repeating pattern and would be captured by the median model. Accordingly, time-frequency bins with large deviations at period p would constitute a non-repeating pattern and would be removed by the median model.

The median is preferred to the geometrical mean originally used in [29] because it was found to lead to a better discrimination between repeating and non-repeating patterns. Note that the use of the median is the reason why we chose to estimate the repeating period in the first 1/3 of the stable portion of the beat spectrum, because we need at least three segments to define a reasonable median. The segmentation of the mixture spectrogram V and the computation of the repeating segment model S are illustrated in the middle row of Fig. 1.

C. Repeating Patterns Extraction

Once the repeating segment model S is calculated, we use it to derive a repeating spectrogram model W, by taking the element-wise minimum between S and each of the r segments of the spectrogram V, as exemplified in the bottom row of Fig. 1. As noted in [30], if we assume that the non-negative spectrogram V is the sum of a non-negative repeating spectrogram W and a non-negative non-repeating spectrogram $V - W$, then we must have $W \leq V$, element-wise, hence the use of the minimum function. The calculation of the repeating spectrogram model W is shown in (3).

$$W(i, l+(k-1)p) = \min\{S(i,l), V(i, l+(k-1)p)\} \quad \text{for } i = 1\ldots n,\ l = 1\ldots p,\ \text{and } k = 1\ldots r. \tag{3}$$

Once the repeating spectrogram model W is calculated, we use it to derive a soft time-frequency mask M, by normalizing W by V, element-wise. The idea is that time-frequency bins that are likely to repeat at period p in V will have values near 1 in M and will be weighted toward the repeating background, and time-frequency bins that are not likely to repeat at period p in V would have values near 0 in M and would be weighted toward the non-repeating foreground. The calculation of the soft time-frequency mask M is shown in (4).

$$M(i,j) = \frac{M(i,j)}{V(i,j)} \quad \text{with } M(i,j) \in [0,1] \quad \text{for } i = 1\ldots n \text{ (frequency) and } j = 1\ldots m \text{ (time).} \tag{4}$$

The time-frequency mask M is then symmetrized and applied to the STFT X of the mixture x. The estimated music signal is obtained by inverting the resulting STFT into the time

domain. The estimated voice signal is obtained by simply subtracting the time-domain music signal from the mixture signal x. The derivation of the repeating spectrogram model W and the building of the soft time-frequency mask M are illustrated in the bottom row of Fig. 1.

We could also further derive a binary time-frequency mask by forcing time-frequency bins in M with values above a certain threshold $t \in [0, 1]$ to 1, while the rest is forced to 0. Our experiments actually showed that the estimates sound perceptually better when using a soft time-frequency mask.

III. Music/Voice Separation on Song Clips 1

In this section, we evaluate REPET on a data set of 1,000 song clips, compared with a recent competitive singing voice separation method. We first introduce the data set (Section III-A) and the competitive method (Section III-B). We then present the performance measures (Section III-C). We finally present the experimental settings (Section III-D) and the comparative results (Section III-E).

A. Data Set 1

Hsu *et al.* proposed a data set called MIR-1K[1]. The data set consists of 1,000 song clips in the form of split stereo WAVE files sampled at 16 kHz, extracted from 110 karaoke Chinese pop songs, performed mostly by amateurs, with the music and voice recorded separately on the left and right channels, respectively. The duration of the clips ranges from 4 to 13 seconds. The data set also includes manual annotations of the pitch contours, indices of the vocal/non-vocal frames, indices and types of the unvoiced vocal frames, and lyrics [23].

Following the framework adopted by Hsu *et al.* in [23], we used the 1,000 song clips of the MIR-1K data set to create three sets of 1,000 mixtures. For each clip, we mixed the music and the voice components into a monaural mixture using three different "voice-to-music" ratios: -5 dB (music is louder), 0 dB (same original level), and 5 dB (voice is louder).

B. Competitive Method 1

Hsu *et al.* proposed a singing voice separation system based on a pitch-based inference technique [23] (see Section I-B). They used the predominant pitch estimation algorithm of Dressler, which got the best overall accuracies for the task of audio melody extraction in the Music Information Retrieval Evaluation eXchange (MIREX) of 2005, 2006, and 2009[2].

C. Performance Measures

To measure performance in source separation, Févotte *et al.* designed the *BSS_EVAL toolbox*[3]. The toolbox proposes a set of measures that intend to quantify the quality of the separation between a source and its estimate. The principle is to decompose an estimate $\hat{s}$ of a source s as follows:

$$\hat{s}(t) = s_{target}(t) + e_{interf}(t) + e_{noise}(t) + e_{artif}(t) \quad (5)$$

where s_{target} is an allowed distortion of source s, and e_{interf}, e_{noise}, and e_{artif} represent respectively the interferences of the unwanted sources, the perturbation noise, and the artifacts introduced by the separation algorithm [31]. We do not assume any perturbation noise, so we can drop the e_{noise} term.

The following performance measures can then be defined: Source-to-Distortion Ratio (SDR), Source-to-Interferences Ratio (SIR) and Sources-to-Artifacts Ratio (SAR).

$$SDR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{interf} + e_{artif}\|^2}\right) \quad (6)$$

$$SIR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{interf}\|^2}\right) \quad (7)$$

$$SAR = 10\log_{10}\left(\frac{\|s_{target} + e_{interf}\|^2}{\|e_{artif}\|^2}\right). \quad (8)$$

Higher values of SDR, SIR, and SAR suggest better separation. We chose those measures because they are widely known and used, and also because they have been shown to be well correlated with human assessments of signal quality [32].

Following the framework adopted by Hsu *et al.* in [23], we then computed the Normalized SDR (NSDR) which measures the improvement of the SDR between the estimate $\hat{s}$ of a source s and the mixture x, and the Global NSDR (GNSDR) which measures the overall separation performance, by taking the mean of the NSDRs over all the mixtures x_k of a given mixture set, weighted by their length w_k. Higher values of NSDR and GNSDR suggest better separation.

$$NSDR(\hat{s}, s, x) = SDR(\hat{s}, s) \quad SDR(x, s) \quad (9)$$

$$GNSDR = \frac{\left(\sum_k w_k \, N\bar{S}DR(\hat{s}_k, s_k, x_k)\right)}{\sum_k w_k}. \quad (10)$$

D. Experimental Settings

We calculated the STFT X of all the mixtures for the three mixture sets (-5, 0, and 5 dB), using half-overlapping Hamming windows of $N = 1024$ samples, corresponding to 64 milliseconds at 16 kHz. The repeating period p was automatically estimated using Algorithm 1. We derived only a soft time-frequency mask as described in (4), because pilot experiments showed that the estimates sound perceptually better in that case. In addition, we applied a high-pass filtering with a cutoff frequency of 100 Hz on the voice estimates. This means that all the energy under 100 Hz in the voice estimates was transferred to the corresponding music estimates. The rationale is that singing voice rarely happens below 100 Hz.

We compared REPET with the best automatic version of Hsu's system, i.e., with estimated pitch, computer-detected unvoiced frames, and singing voice enhancement [23], and also with the initial version of REPET with binary masking used in [29]. Since Hsu *et al.* reported the results only for the voice estimates in [23], we evaluated REPET here only for the extraction of the voice component.

Following the framework adopted by Hsu *et al.* in [23], we calculated the NSDR for all the voice estimates and measured

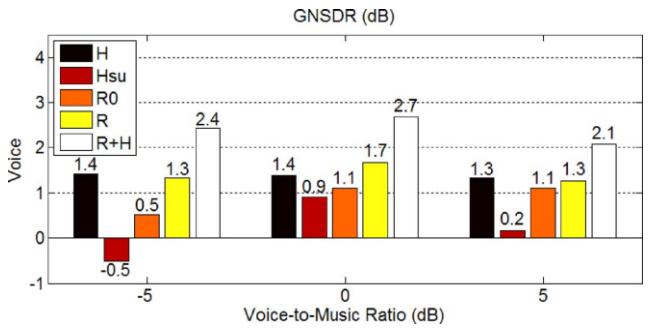

Fig. 2. Separation performance via the GNSDR in dB, for the voice component, at voice-to-music ratios of , 0, and 5 dB, from left to right, using only a high-pass filtering (, *black*), Hsu's system (, *dark color*), the initial REPET with binary masking (, *medium color*), REPET (with soft masking) (, *light color*), and REPET plus high-pass filtering (, *white*). Higher values are better.

the separation performance for the voice component by computing the GNSDR for each of the three mixture sets. We also computed the NSDRs and GNSDRs directly from the mixtures after a simple high-pass filtering of 100 Hz.

E. Comparative Results

Fig. 2 shows the separation performance via the GNSDR in dB, for the voice component, at voice-to-music ratios of 5, 0, and 5 dB. From left to right, the black bars represent using only a high-pass filtering on the mixtures (H). The dark-colored bars represent Hsu's system (Hsu). The medium-colored bars represent the initial REPET with binary masking ($R0$). The light-colored bars represent REPET (with soft masking) (R). The white bars represent REPET plus high-pass filtering ($R+H$). Higher values are better.

As we can see in Fig. 2, a simple high-pass filtering on the mixtures can give high GNSDRs for the voice estimates, although the GNSDRs for the music estimates (not shown here) are much lower in comparison. REPET gives higher GNSDRs for the voice estimates compared with Hsu's system, and the initial REPET, while giving satisfactory GNSDRs for the music estimates (not shown here). Finally, a high-pass filtering on the voice estimates of REPET is shown to boost the GNSDRs. Note that in [29], the algorithm for estimating the repeating period was tuned for the initial REPET to lead to the best voice estimates, regardless of the separation performance for the music estimates, while here Algorithm 1 is tuned for REPET to lead to the best music *and* voice estimates.

A series of multiple comparison statistical tests showed that, for the voice component, $R+H$ gives statistically the best NSDR, for all the three voice-to-music ratios. R gives statistically better NSDR compared with H, except at -5 dB where there is no statistically significant difference. For the music component, $R+H$ still gives statistically the best NSDR, and H gives statistically the worst NSDR, considerably worse than with the voice component, for all the three voice-to-music ratios. Since Hsu *et al.* reported their results only using the GNSDR, which is a weighted mean, we were not able to perform a statistical comparison with Hsu's system. We used ANOVA when the compared distributions were all normal, and a Kruskal-Wallis test when at least one of the compared distributions was not normal. We used a Jarque-Bera normality test to determine if a distribution was normal or not.

The high NSDRs and GNSDRs observed for H for the voice component are probably due to the fact that, although not leading to good separation results, using a high-pass filtering of 100 Hz on the mixtures still yields some improvement of the SDR between the voice estimates and the mixtures, since singing voice rarely happens below 100 Hz. However, this also means leaving only the energy below 100 Hz for the music estimates, which obviously yields very bad NSDRs and GNSDRs, since music does not happen only below 100 Hz.

In this section, we showed that REPET can compete with a recent singing voice separation method. However, there might be some limitations with this evaluation. First, Hsu *et al.* reported their results only using the GNSDR. The GNSDR is a single value that intends to measure the separation performance of a whole data set of 1,000 mixtures, which makes us wonder if it is actually reliable, especially given the high values obtained when using a simple high-pass filtering on the mixtures. Also, the GNSDR is a weighted mean, which prevents us for doing a comparison with the competitive method, because no proper statistical analysis is possible.

Then, Hsu *et al.* reported the results only for the voice estimates. We showed that reporting the results for one component only is not sufficient to assess the potential of a separation algorithm. Also, this prevents us for comparing our music estimates. In the next section, we therefore propose to conduct a new evaluation, comparing REPET with another recent competitive method, for the separation of both the music and voice components, using the standard SDR, SIR, and SAR.

IV. Music/Voice Separation on Song Clips 2

In this section, we evaluate REPET on the same data set of song clips, compared with another competitive music/voice separation. We first introduce the new competitive method (Section IV-A). We then present the experimental settings (Section IV-B) and the comparative results (Section IV-C). We finally investigate potential improvements (Section IV-D) and analyze the interactions between length, repetitions, and performance in REPET (Section IV-E).

A. Competitive Method 2

Durrieu *et al.* proposed a music/voice separation method based on a source/filter modeling [24] (see Section I-B). Given a WAVE file as an input, the program[4] outputs four WAVE files: the accompaniment and lead estimates, with and without unvoiced lead estimation. We used an analysis window of 64 milliseconds, an analysis Fourier size of $N = 1024$ samples, a step size of 32 milliseconds, and a number of 30 iterations.

B. Experimental Settings

For the evaluation, we used the MIR-1K data set, with the three mixture sets (see Section III-A). To measure performance in source separation, we used the standard SDR, SIR, and SAR (see Section III-C). For the parameterization of REPET, we

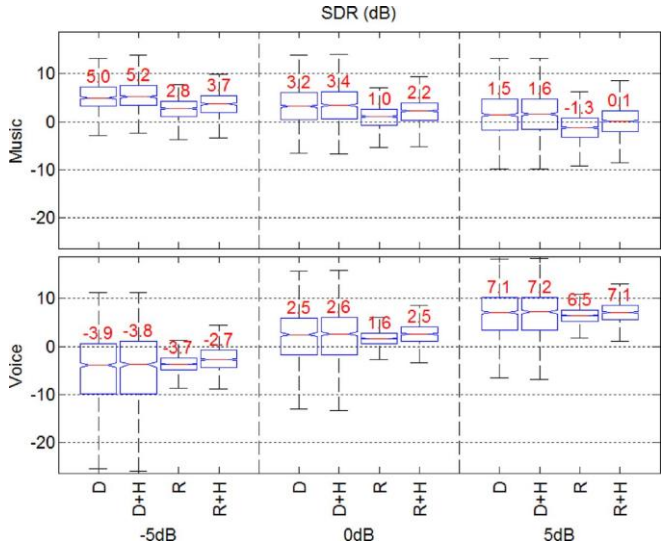

Fig. 3. Separation performance via the SDR in dB, for the music (top plot) and voice (bottom plot) components, at voice-to-music ratios of (left column), 0 (middle column), and 5 dB (right column), using Durrieu's system (D), Durrieu's system plus high-pass filtering (D + H), REPET (R), and REPET plus high-pass filtering (R + H). Outliers are not shown. Median values are displayed. Higher values are better.

used the same settings used in the previous evaluation (see Section III-D).

We compared REPET with Durrieu's system enhanced with the unvoiced lead estimation [24]. We also applied a high-pass filtering of 100 Hz on the voice estimates for both methods.

C. Comparative Results

Fig. 3 shows the separation performance via the SDR in dB, for the music (top plot) and voice (bottom plot) components, at voice-to-music ratios of -5 (left column), 0 (middle column), and 5 dB (right column). In each column, from left to right, the first box represents Durrieu's system. (D). The second box represents Durrieu's system plus high-pass filtering ($D + H$). The third box represents REPET (R). The fourth box represents REPET plus high-pass filtering ($R + H$). The horizontal line in each box represents the median of the distribution, whose value is displayed above the box. Outliers are not shown. Higher values are better.

As we can see in Fig. 3, a high-pass filtering on the voice estimates of Durrieu's system increases the SDR, but also the SIR (not shown here), for both the music and voice components, and for all the three voice-to-music ratios. While it also increases the SAR for the music component, it however decreases the SAR for the voice component (not shown here). The same behavior is observed for REPET. A series of multiple comparison statistical tests showed that the improvement for Durrieu's system is statistically significant only for the SAR for the music component and the SIR for the voice component. The improvement for REPET is statistically significant in all cases, except for the SAR for the voice component where a statistically significant decrease is observed. This suggests that the high-pass filtering helps REPET more than it helps Durrieu's system.

As we can also see in Fig. 3, compared with Durrieu's system, with or without high-pass filtering, REPET gives lower SDR for the music component, for all the three voice-to-music ratios. The same results are observed for the SIR for the voice component and the SAR for the music component. With high-pass filtering, REPET gives similar SDR for the voice component, and even higher SDR at -5 dB. REPET gives also higher SIR for the music component at -5 dB, and higher SAR for the voice component for all the three voice-to-music ratios. This suggests that, although Durrieu's system is better at removing the vocal interference from the music, it also introduces more artifacts in the music estimates. REPET gets also better than Durrieu's system at removing the musical interference from the voice as the music gets louder. This makes sense since REPET models the musical background. A series of multiple comparison statistical tests showed that those results were statistically significant in all cases.

Durrieu's system shows also larger statistical dispersions, and this for all the three performance measures, for both the music and voice components, and for all the three voice-to-music ratios. This suggests that, while being sometimes much better than REPET, it is also sometimes much worse.

The average computation time for REPET, over all the mixtures and all of the three mixture sets, was 0.016 second for 1 second of mixture, when implemented in Matlab. The average computation time for Durrieu's system was 3.863 seconds for 1 second of mixture, when implemented in Python. Both algorithms ran on the same PC with Intel Core2 Quad CPU of 2.66 GHz and 6 GB of RAM. This shows that, in addition to being competitive with a recent music/voice separation method, REPET is also much faster.

D. Potential Improvements

We now investigate potential improvements to REPET. First, we consider a post-processing of the outputs, by using a high-pass filtering of 100 Hz on the voice estimates (see above). This can be done automatically without any additional information. Then, we consider an optimal parameterization of the algorithm, by selecting the repeating period that leads to the best mean SDR between music and voice estimates. This shows the maximal improvement possible given the use of an ideal repeating period finder. Finally, we consider prior information about the inputs, by using the indices of the vocal frames. This shows the maximal improvement possible given the use of an ideal vocal/non-vocal discriminator.

Fig. 4 shows the separation performance via the SDR in dB, for the music (top plot) and voice (bottom plot) components, at voice-to-music ratios of -5 (left column), 0 (middle column), and 5 dB (right column). In each column, from left to right, the first box represents REPET (R). The second box represents REPET, plus high-pass filtering ($R + H$). The third box represents REPET, plus high-pass filtering, plus the best repeating period ($R + H + P$). The fourth box represents REPET, plus high-pass filtering, plus the best repeating period, plus the indices of the vocal frames ($R + H + P + V$).

As we can see in Fig. 4, the high-pass filtering, the best repeating period, and the indices of the vocal frames successively improve the SDR, for both the music and voice components, and for all the three voice-to-music ratios. A similar pattern is also observed for the SIR and SAR (not shown here), for both

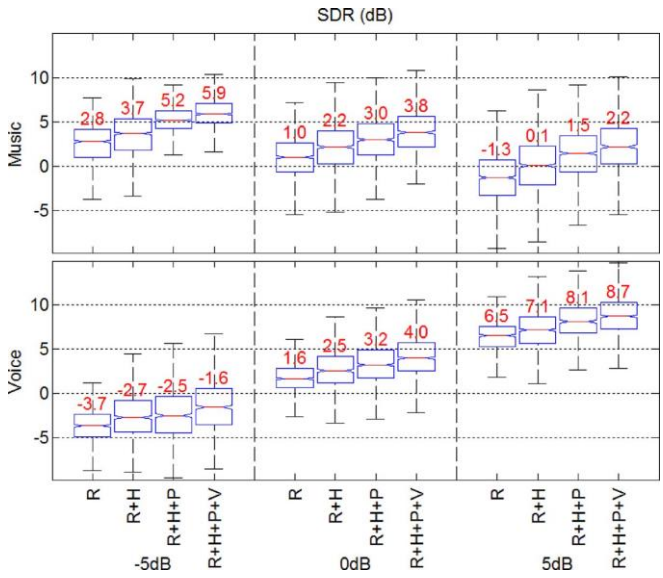

Fig. 4. Separation performance via the SDR in dB, for the music (top plot) and voice (bottom plot) components, at voice-to-music ratios of (left column), 0 (middle column), and 5 dB (right column), using REPET (R), then enhanced with a high-pass filtering (R + H), further enhanced with the best repeating period (R + H + P), and finally enhanced with the indices of the vocal frames (R + H + P + V).

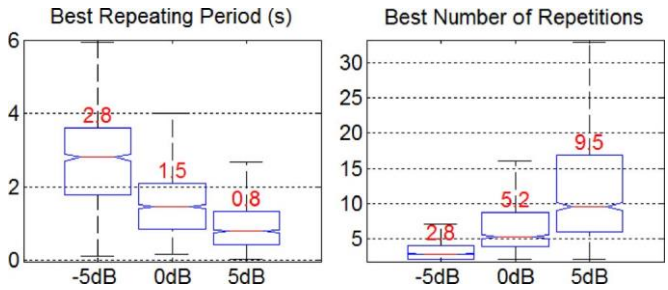

Fig. 5. Distributions for the best repeating period in seconds (left plot) and the corresponding number of repetitions (right plot) for REPET, at voice-to-music ratios of 5, 0, and 5 dB.

the music and voice components, and for all the three voice-to-music ratios, except for the SAR for the voice component. This suggests that there is still room for improvement for REPET. A series of multiple comparison statistical tests showed that those results are statistically significant in all cases, except for the SAR for the voice component where a statistically significant decrease is observed.

E. Interactions Between Length, Repetitions, and Performance

Fig. 5 shows the distributions of the best repeating period in seconds (left plot) and the corresponding number of repetitions (right plot) for REPET, at voice-to-music ratios of 5, 0, and 5 dB. As we can see, as the voice-to-music ratio gets larger, the best repeating period gets smaller, and the number of repetitions gets larger. This suggests that, as the voice gets louder, REPET needs more repetitions to derive effective repeating segment models, which constrains REPET to dig into the finer repeating structure (e.g., at the beat level).

In addition, we found that there is no correlation between the mixture length and the best number of repetitions, or the performance measures, and this for all the three voice-to-music ratios. This suggests that the mixture length has no influence on REPET here. We also found that, as the voice-to-music ratio gets smaller, a positive correlation appears between the best number of repetitions and the performance measures, given the SIR for the music component, and the SDR and SIR for the voice component, while a negative correlation appears given the SAR for the voice component. This suggests that, as the music gets louder, a larger number of repetitions means a reduction of the interferences in the music and voice estimates, but also an increase of the artifacts in the voice estimates. We used the Pearson product-moment correlation coefficient.

In this section, we showed that REPET can compete with another recent music/voice separation method. However, there might also be some limitations with this evaluation. First, the MIR-1K data set was created from karaoke pop songs. The recordings are not of great quality; some vocals are still present in some of the accompaniments. Also, it could be interesting to evaluate REPET on real-world recordings.

Then, the MIR-1K data set is composed of very short clips. REPET needs sufficiently long excerpts to derive good repeating segment models. Also, it could be interesting to evaluate REPET on full-track songs. In the next section, we propose to conduct a new evaluation, analyzing the applicability of REPET on a new data set of full-track real-world songs.

V. Music/Voice Separation on Full Songs

In this section, we evaluate the applicability of REPET on a new data set of 14 full-track real-world songs. We first introduce the new data set (Section V-A). We then propose a simple procedure to extend REPET to longer pieces (Section V-B). We then present the experimental settings (Section V-C). We then analyze the interactions between length, repetitions, and performance (Section V-D). We finally show some comparative results (Section V-E).

A. Data Set 2

The new data set consists of 14 full-track real-world songs, in the form of split stereo WAVE files sampled at 44.1 kHz, with the music and voice recorded separately on the left and right channels, respectively. These 14 stereo sources were created from live-in-the-studio recordings released by *The Beach Boys*, where some of the accompaniments and vocals were made available as split stereo tracks[5] and separated tracks[6]. The duration of the songs ranges from 2'05" to 3'10." For each song, we mixed the music and voice components into a monaural mixture at voice-to-music ratio of 0 dB only.

B. Extended REPET

For excerpts with a relatively stable repeating background (e.g., 10 second verse), we showed that REPET can be successfully applied for music/voice separation (see Sections III and IV). For full-track songs, the repeating background is likely to show variations over time (e.g., verse followed by chorus).

We could extend REPET to full-track songs by applying the algorithm to individual sections where the repeating background

[5]Good Vibrations: Thirty Years of The Beach Boys, 1993

[6]The Pet Sounds Sessions, 1997

is stable (e.g., verse/chorus). This could be done by first performing an audio segmentation of the song. For example, an interesting work could be that of Weiss *et al.* [33], who proposed to automatically identify repeated patterns in music using a sparse shift-invariant PLCA, and showed how such analysis can be applied for audio segmentation (see also Section I-A).

Recently, Liutkus *et al.* proposed to adapt the REPET algorithm along time to handle variations in the repeating background [34]. The method first tracks local periods of the repeating structure, then models local estimates of the repeating background, and finally extracts the repeating patterns.

Instead, we propose a very simple procedure to extend REPET to longer pieces. We simply apply the algorithm to local windows of the signal over time. Given a window size and an overlap percentage, we successively extract the local repeating backgrounds using REPET. We then reconstruct the whole repeating background via overlap-add, after windowing the overlapping parts to prevent from reconstruction artifacts.

C. Experimental Settings

We evaluated this extended REPET on the Beach Boys data set, using different window sizes (2.5, 5, 10, 20, and 40 seconds), and overlap percentages (0, 25, 50, and 75%). We calculated the STFT X for each window in a mixture, using half-overlapping Hamming windows of $N = 2048$ samples, corresponding to 46.4 milliseconds at 44.1 kHz. The repeating period p was automatically estimated using Algorithm 1. We also applied REPET on the full mixtures without windowing.

We compared this extended REPET with Durrieu's system enhanced with the unvoiced lead estimation [24]. We used an analysis window of 46.4 milliseconds, an analysis Fourier size of $N = 2048$ samples, a step size of 23.2 milliseconds, and a number of 30 iterations. We also applied a high-pass filtering of 100 Hz on the voice estimates for both methods, and use the best repeating period for REPET.

D. Interactions Between Length, Repetitions, and Performance

Fig. 6 shows the separation performance via the SDR in dB, for the music (left plot) and voice (right plot) components, using the extended REPET with windows of *2.5*, *5*, *10*, *20*, and *40* seconds, and overlap of 75%, and the full REPET without windowing (*full*). We evaluated the extended REPET for overlap of 75% only, because our experiments showed that overall the performance measures were higher in that case, for both the music and voice components, although a series of multiple comparison statistical tests showed that there was no statistically significant difference between the overlaps.

As we can see in Fig. 6, there is an overall bell-shaped curve, with the extended REPET with window of 10 seconds having the highest SDR, and the full REPET having the lowest SDR. A similar curve is also observed for the SIR and SAR (not shown here), for both the music and voice components, except for the SAR for the voice component. This suggests that there is a trade-off for the window size in REPET. If the window is too long, the repetitions will not be sufficiently stable; if the window is too short, there will not be sufficient repetitions. This is closely related with the time/frequency trade-off of the STFT. A series of multiple comparison statistical tests showed that

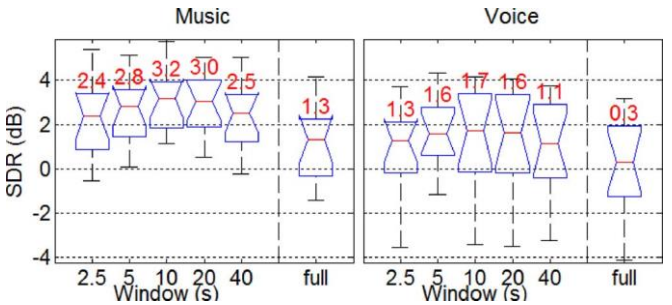

Fig. 6. Separation performance via the SDR in dB, for the music (left plot) and voice (right plot) components, using the extended REPET with windows of *2.5*, *5*, *10*, *20*, and *40* seconds, and overlap of 75%, and the full REPET without windowing (*full*).

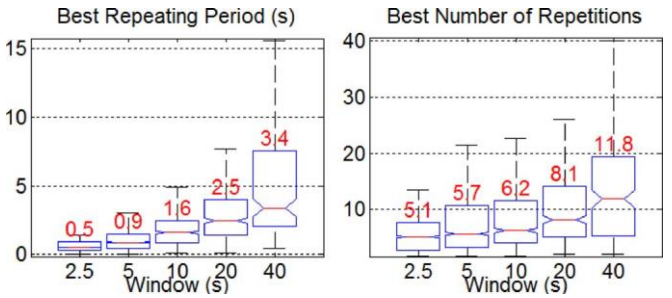

Fig. 7. Distributions for the best repeating period in seconds (left plot) and the corresponding number of repeating segments (right plot), in one window, for the extended REPET with windows of *2.5*, *5*, *10*, *20*, and *40* seconds, and overlap of 75%.

there is overall no statistically significant difference between the windows.

Fig. 7 shows the distributions for the best repeating period in seconds (left plot), and the corresponding number of repetitions (right plot), in one window, for the extended REPET with windows of *2.5*, *5*, *10*, *20*, and *40* seconds, and overlap of 75%. As we can see, REPET has a minimum median of 5.1 repetitions. This is line with the recent findings that the performance of the human auditory system in segregating the same embedded repeating sound in different mixtures asymptotes with about five mixtures [28].

In addition, we found that, as the window size gets larger, the SDR, SIR, and SAR for the music component decrease from positive correlations between the best number of repetitions and the performance measures to negative correlations, while they increase for the voice component from no correlation to positive correlations. This suggests that a smaller repeating period is likely to give better voice estimates, while a larger repeating period is likely to give better music estimates.

E. Comparative Results

Fig. 8 shows the separation performance via the SDR in dB, for the music (left plot) and voice (right plot) components. In each plot, from left to right, the first box represents Durrieu's system (D). The second box represent Durrieu's system plus high-pass filtering ($D + H$). The third box represents the extended REPET with window of 10 seconds and overlap of 75% (R). The fourth box represents the extended REPET plus high-pass filtering ($R + H$). The fourth box represents the extended

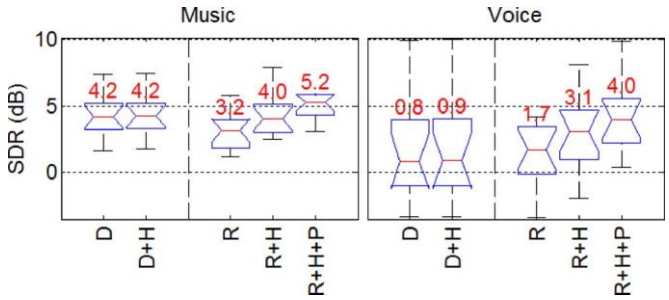

Fig. 8. Separation performance via the SDR in dB, for the music (left plot) and voice (right plot) components, using Durrieu's system (D), Durrieu's system plus high-pass filtering (D + H), the extended REPET with window of 10 seconds and overlap of 75% (R), the extended REPET plus high-pass filtering (R + H), and the extended REPET plus high-pass filtering, plus the best repeating period (R + H + P).

REPET, plus high-pass filtering, plus the best repeating period ($R + H + P$).

As we can see in Fig. 8, a high-pass filtering on the voice estimates of Durrieu's system increases the SDR, and also the SIR (not shown here), for both the music and voice components. While it also increases the SAR for the music component, it however decreases the SAR for the voice component (not shown here). The same behavior is observed for the extended REPET. The best repeating period further improves the SDR, and also the SAR, for both the music and voice components. While it also increases the SIR for the music component, it however decreases the SIR for the voice component. A series of multiple comparison statistical tests showed that the improvements for Durrieu's system are not statistically significant. The improvement for the extended REPET are statistically significant only for the SDR for the music component where a statistically significant increase is observed between R and $R + H + P$, and for the SAR for the voice component where a statistically significant decrease is observed between R and $R + H$.

As we can also see in Fig. 8, compared with Durrieu's system, with or without high-pass filtering, REPET gives higher SDR, and also SAR, for the music component, when enhanced with both a high-pass filtering and the best repeating period. For the voice component, REPET gives higher SDR, and also SAR, in all cases. REPET gives higher SIR for the music component when only enhanced with a high-pass filtering. A series of multiple comparison statistical tests showed that those results were actually not statistically significant.

The average computation time for the extended REPET with a window of 10 seconds and an overlap of 75%, over all the mixtures of the Beach Boys data set, was 0.212 second for 1 second of mixture. The average computation time for Durrieu's system was 7.556 seconds for 1 second of mixture. These results show that REPET is applicable on full-track real-world songs, competing with a recent music/voice separation method.

VI. Melody Extraction

In this section, we evaluate REPET as a preprocessor for two pitch detection algorithms to improve melody extraction. We first introduce the two pitch detection algorithms (Section VI-A). We then present the performance measures (Section VI-B). We finally show the extraction results (Section VI-C).

A. Pitch Detection Algorithms

We have shown that REPET can be successfully applied for music/voice separation. We now show that REPET can consequently improve melody extraction, by using it to first separate the repeating background, and then applying a pitch detection algorithm on the voice estimate to extract the pitch contour. We employ two different pitch detection algorithms: the well-known single fundamental frequency (F_0) estimator YIN proposed by de Cheveigné *et al.* in [35], and the more recent *multiple* F_0 estimator proposed by Klapuri in [36].

For the evaluation, we used the MIR-1K data set, with the three derived mixture sets. As ground truth, we used the provided manual annotated pitch contours. The frame size corresponds to 40 milliseconds with half-overlapping, and the pitch values are in semitone, encoded as MIDI numbers. Values of 0 represent frames where no voice is present.

YIN is an F_0 estimator designed for speech and music, based on the autocorrelation method [35]. Given a sampled signal as an input, the program[7] outputs a vector of F_0 estimates in octaves, a vector of aperiodicity measures, and a vector of powers. We fixed the range of F_0 candidates between 80 and 1280 Hz. We used a frame size of 40 milliseconds with half-overlapping. By default, YIN outputs a pitch estimate for every frame. We can however discard unlikely pitch estimates, i.e., those that show too much aperiodicity and not enough power. Pilot experiments showed that thresholds of 0.5 for the aperiodicity and 0.001 for the power (after normalization by the maximum) lead to good pitch estimates.

Klapuri proposed a multiple F_0 estimator designed for *polyphonic* music signals, based on an iterative estimation and cancellation of the multiple F_0s [36]. Given a sampled signal as an input, the program outputs a vector of F_0 estimates in Hz, and a vector of F_0 saliences. We fixed the range of F_0 candidates between 80 and 1280 Hz. We used a frame size of 2048 samples and a hop size of 882 samples. By default, Klapuri's system outputs a pitch estimate for every frame. We can however discard unlikely pitch estimates, i.e., those that do not show sufficient salience. Pilot experiments showed that a threshold of 0.3 for the salience (after normalization by the maximum) leads to good pitch estimates.

B. Performance Measures

To measure performance in pitch estimation, we used the *precision*, *recall*, and F*-measure*. We define *true positive (tp)* to be the number of correctly estimated pitch values compared with the ground truth pitch contour, *false positive (fp)* the number of incorrectly estimated pitch values, and *false negative (fn)* the number of incorrectly estimated non-pitch values. A pitch estimate was treated as correct if the absolute difference from the ground truth was less than 1 semitone.

We then define *precision* (P) to be the percentage of estimated pitch values that are correct, *recall* (R) the percentage

of correct pitch values that are estimated, and *F-measure* the harmonic mean between P and R. Higher values of precision, recall, and F-measure suggest better pitch estimation.

$$precision = \frac{tp}{(tp + f(k} = P \tag{11}$$

$$recall = \frac{tp}{(tp + fn)} = R \tag{12}$$

$$F\text{-}measure = \frac{2\,(P \times R)}{(P + R)}. \tag{13}$$

C. Extraction Results

We extracted the pitch contours from the voice estimates obtained from REPET, including the potential enhancements (see Section IV-D), using YIN and Klapuri's system. We also extracted the pitch contours from the mixtures and the voice sources to serve, respectively, as a lower-bound and upper-bound on the performance in pitch estimation. Performance in pitch estimation was measured by using the precision, recall, and F -measure, in comparison with the ground truth pitch contours.

Fig. 9 shows the melody extraction performance via the F-measure, at voice-to-music ratios of -5 (left column), 0 (middle column), and 5 dB (right column), using YIN (top plot) and Klapuri's system (bottom plot). In each column, from left to right, the first box represents the results from the mixtures (*mixtures*). The second box represents the results using REPET plus high-pass filtering ($R + H$). This represents the improvement from applying an automatic REPET. The third box represents the results using REPET plus high-pass filtering, plus the best repeating period and the indices of the vocal frames ($R + H + P + V$). This represents the improvement from applying an ideal REPET. The fourth box represents the results from the voice sources (*voices*).

As we can see in Fig. 9, compared with extracting the pitch directly from the mixtures, using REPET plus high-pass filtering to first extract the voice estimates improves the F-measure, for all the three voice-to-music ratios, for both YIN and Klapuri's system. The best repeating period and the indices of the vocal frames further improves the F-measure. A similar pattern is also observed for the precision (not shown here), for all the three voice-to-music ratios, for both YIN and Klapuri's system. As for the recall (not shown here), in the case of YIN, while using REPET plus high-pass filtering improves the results, the additional enhancements do not further improve them. In the case of Klapuri's system, a decrease is actually observed for the recall.

A series of multiple comparison statistical tests showed that those results are statistically significant in all cases, for both the F-measure and the precision. As for the recall, in the case of YIN, using REPET plus high-pass filtering is shown to statistically improve the results, however a statistically significant decrease is then observed when further adding the best repeating period and the indices of the vocal frames. In the case of Klapuri's system, a statistically significant decrease is actually observed for the recall. These results also confirm that overall there is still room for improvement for REPET.

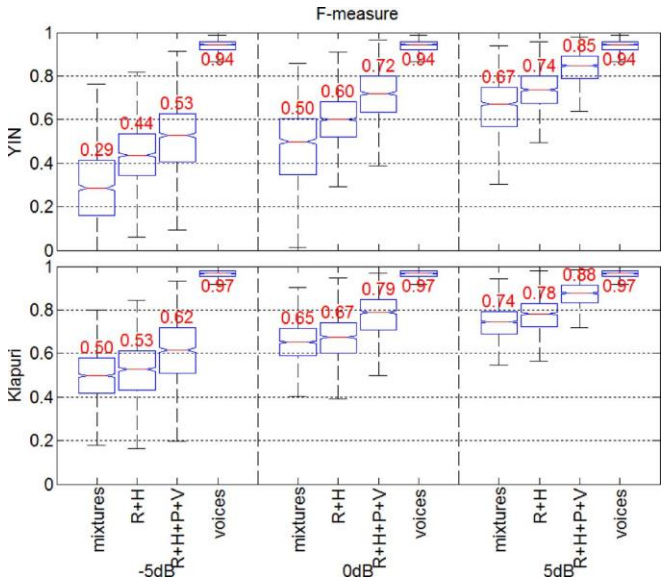

Fig. 9. Melody extraction performance via the -measure, at voice-to-music ratios of (left column), 0 (middle column), and 5 dB (right column), using YIN (top plot) and Klapuri's system (bottom plot), on the mixtures (*mixtures*), on the voice estimates of REPET plus high-pass filtering (R + H), then enhanced with the best repeating period and the indices of the vocal frames (R + H + P + V), and on the voice sources (*voices*).

VII. Conclusion

In this work, we have presented the REpeating Pattern Extraction Technique (REPET), a novel and simple approach for separating the repeating background from the non-repeating foreground in a mixture. The basic idea is to identify the periodically repeating segments in the audio, compare them to a repeating segment model derived from them, and extract the repeating patterns via time-frequency masking.

Experiments on a data set of 1,000 song clips showed that REPET can be efficiently applied for music/voice separation, competing with two state-of-the-art approaches, while still showing room for improvement. More experiments on a data set of 14 full-track real-world songs showed that REPET is robust to real-world recordings and can be easily extended to full-track songs. Further experiments showed that REPET can also be used as a preprocessor to pitch detection algorithms to improve melody extraction.

In addition, more information about REPET, including the source code and audio examples, can be found online[8].

Acknowledgment

The authors would like to thank C.-L. Hsu for providing the results of his singing voice separation system, J.-L. Durrieu for helping with the code for his music/voice separation system, and A. Klapuri for providing the code for his multiple F_0 estimator. We also would like to thank A. Liutkus and his colleagues from Telecom Paristech for their fruitful discussions, and our colleagues from the Interactive Audio Lab, M. Cartwright, Z. Duan, J. Han, and D. Little for their thoughtful comments. Finally, we would like to thank the reviewers for their helpful reviews.

REFERENCES

[1] H. Schenker, *Harmony*. Chicago, IL: Univ. of Chicago Press, 1954.

[2] N. Ruwet and M. Everist, "Methods of analysis in musicology," *Music Anal.*, vol. 6, no. 1/2, pp. 3–9+11–36, Mar.-Jul. 1987.

[3] A. Ockelford, *Repetition in Music: Theoretical and Metatheoretical Perspectives*. Farnham, U.K.: Ashgate, 2005, vol. 13, Royal Musical Association Monographs.

[4] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. 7th ACM Int. Conf. Multimedia (Part 1)*, Orlando, FL, Oct.-Nov. 30–05, 1999, pp. 77–80.

[5] M. Cooper and J. Foote, "Automatic music summarization via similarity analysis," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 13–17, 2002, pp. 81–85.

[6] A. Pikrakis, I. Antonopoulos, and S. Theodoridis, "Music meter and tempo tracking from raw polyphonic audio," in *Proc. 9th Int. Conf. Music Inf. Retrieval*, Barcelona, Spain, Oct. 10–14, 2008.

[7] G. Peeters, "Deriving musical structures from signal analysis for music audio summary generation: "Sequence" and "state" approach," in *Computer Music Modeling and Retrieval*, U. Wiil, Ed. Berlin/Heidelberg, Germany: Springer, 2004, vol. 2771, Lecture Notes in Computer Science, pp. 169–185.

[8] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. IEEE Int. Conf. Multimedia and Expo*, New York, Jul.-Aug. 30–02, 2000, vol. 1, pp. 452–455.

[9] J. Foote and S. Uchihashi, "The beat spectrum: A new approach to rhythm analysis," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Tokyo, Japan, Aug. 22–25, 2001, pp. 881–884.

[10] M. A. Bartsch, "To catch a chorus using chroma-based representations for audio thumbnailing," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, New Paltz, NY, Oct. 21–24, 2001, pp. 15–18.

[11] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *J. New Music Res.*, vol. 32, no. 2, pp. 153–164, 2003.

[12] K. Jensen, "Multiple scale music segmentation using rhythm, timbre, and harmony," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, pp. 1–11, Jan. 2010.

[13] R. B. Dannenberg, "Listening to "Naima": An automated structural analysis of music from recorded audio," in *Proc. Int. Comput. Music Conf.*, Gothenburg, Sweden, Sep. 17–21, 2002, pp. 28–34.

[14] R. B. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. New York: Springer, 2009, pp. 305–331.

[15] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. 11th Int. Soc. Music Inf. Retrieval*, Utrecht, The Netherlands, Aug. 9–13, 2010, pp. 625–636.

[16] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," in *Proc. 6th Int. Conf. Music Inf. Retrieval*, London, U.K., Sep. 11–15, 2005, pp. 337–344.

[17] B. Raj, P. Smaragdis, M. Shashanka, and R. Singh, "Separating a foreground singer from background music," in *Proc. Int. Symp. Frontiers of Res. Speech and Music*, Mysore, India, May 8–9, 2007.

[18] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1564–1578, Jul. 2007.

[19] Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1475–1487, May 2007.

[20] M. Ryynänen, T. Virtanen, J. Paulus, and A. Klapuri, "Accompaniment separation and karaoke application based on automatic melody transcription," in *Proc. IEEE Int. Conf. Multimedia & Expo*, Hannover, Germany, Jun. 23–26, 2008, pp. 1417–1420.

[21] T. Virtanen, A. Mesaros, and M. Ryynänen, "Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music," in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition*, Brisbane, Australia, Sep. 21, 2008, pp. 17–20.

[22] K. Dressler, "An auditory streaming approach on melody extraction," in *Proc. 7th Int. Conf. Music Inf. Retrieval (MIREX Eval.)*, Victoria, BC, Canada, Oct. 8–12, 2006.

[23] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 310–319, Feb. 2010.

[24] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1180–1191, Oct. 2011.

[25] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, The Hague, The Netherlands, Oct. 10–13, 2004, pp. 3099–3104.

[26] K. Yoshii, M. Goto, and H. G. Okuno, "Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates," in *Proc. 5th Int. Conf. Music Inf. Retrieval*, Barcelona, Spain, Oct. 10–14, 2004, pp. 184–191.

[27] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hean, J. R. Zeidler, J. E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc. IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[28] J. H. McDermott, D. Wrobleski, and A. J. Oxenham, "Recovering sound sources from embedded repetition," *Proc. Natural Acad. Sci. United States of Amer.*, vol. 108, no. 3, pp. 1188–1193, Jan. 2011.

[29] Z. Rafii and B. Pardo, "A simple music/voice separation system based on the extraction of the repeating musical structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 22–27, 2011, pp. 221–224.

[30] A. Liutkus and P. Leveau, "Separation of music+effects sound track from several international versions of the same movie," in *Proc. 128th Audio Eng. Soc. Conv.*, London, U.K., May 22–25, 2010.

[31] C. Févotte, R. Gribonval, and E. Vincent, BSS_EVAL Toolbox User Guide IRISA, Rennes, France, 2005, Tech. Rep. 1706.

[32] B. Fox, A. Sabin, B. Pardo, and A. Zopf, "Modeling perceptual similarity of audio signals for blind source separation evaluation," in *Proc. 7th Int. Conf. Ind. Compon. Anal.*, London, U.K., Sep. 09–12, 2007, pp. 454–461.

[33] R. J. Weiss and J. P. Bello, "Identifying repeated patterns in music using sparse convolutive non-negative matrix factorization," in *Proc. 11th Int. Soc. Music Inf. Retrieval*, Utrecht, The Netherlands, Aug. 9–13, 2010.

[34] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard, "Adaptive filtering for music/voice separation exploiting the repeating musical structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 25–30, 2012, pp. 53–56.

[35] A. de Cheveigné, "YIN, A fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.

[36] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. 7th Int. Conf. Music Inf. Retrieval*, Victoria, BC, Canada, Oct. 8–12, 2006, pp. 216–221.

Zafar Rafii (S'11) is a Ph.D. candidate in Electrical Engineering & Computer Science at Northwestern University. He received a Master of Science in Electrical Engineering from Ecole Nationale Superieure de l'Electronique et de ses Applications (ENSEA) in France and from Illinois Institute of Technology (IIT) in Chicago. In France, he worked as a research engineer at Audionamix (aka Mist Technologies). His current research interests are centered around audio analysis and include signal processing, machine learning and cognitive science.

Bryan Pardo (M'07) is an associate professor in the Northwestern University Department of Electrical Engineering and Computer Science. Prof. Pardo received a M. Mus. in Jazz Studies in 2001 and a Ph.D. in Computer Science in 2005, both from the University of Michigan. He has authored over 50 peer-reviewed publications. He is an associate editor for the IEEE TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING. He has developed speech analysis software for the Speech and Hearing department of the Ohio State University, statistical software for SPSS and worked as a machine learning researcher for General Dynamics. While finishing his doctorate, he taught in the Music Department of Madonna University. When he's not programming, writing or teaching, he performs throughout the United States on saxophone and clarinet at venues such as Albion College, the Chicago Cultural Center, the Detroit Concert of Colors, Bloomington Indiana's Lotus Festival and Tucson's Rialto Theatre.