Normalized cuts for predominant melodic source separation

Mathieu Lagrange, Luis Gustavo Martins, Jennifer Murdoch, George Tzanetakis

Normalized Cuts for Predominant Melodic Source Separation

Mathieu Lagrange, *Member, IEEE,* Luis Gustavo Martins, Jennifer Murdoch, *Student Member, IEEE,* and George Tzanetakis, *Member, IEEE*

***Abstract*— The predominant melodic source, frequently the singing voice, is an important component of musical signals. In this paper we describe a method for extracting the predominant source and corresponding melody from "real-world" polyphonic music. The proposed method is inspired by ideas from Computational Auditory Scene Analysis. We formulate predominant melodic source tracking and formation as a graph partitioning problem and solve it using the normalized cut which is a global criterion for segmenting graphs that has been used in Computer Vision. Sinusoidal modeling is used as the underlying representation. A novel harmonicity cue which we term Harmonically Wrapped Peak Similarity is introduced. Experimental results supporting the use of this cue are presented. In addition we show results for automatic melody extraction using the proposed approach.**

***Index Terms*— AUD_CONT,AUD_SSEN (EDICS), music information retrieval, computational auditory scene analysis, sinusoidal modeling, normalized cut**

I. Introduction

THE voice and melodic characteristics of singers are some of the primary features of music to which listeners relate to. Most listeners, independently of their music education, are capable of identifying specific singers even in recordings they have not heard before. In addition, especially in pop and rock music the singing voice carries the main melodic line that can be used to identify a particular song of interest.

Music Information Retrieval (MIR) is a field that has been rapidly evolving over the past few years. It encompasses a wide variety of ideas, algorithms, tools, and systems that have been proposed to handle the increasingly large and varied amounts of musical data available digitally. Typical MIR systems for music signals in audio format represent statistically the entire polyphonic sound mixture [1], [2]. There is some evidence that this approach has reached a "glass ceiling" [3] in terms of retrieval performance.

One obvious direction for further progress is to attempt to individually characterize the different sound sources comprising the polyphonic mixture. The singing voice is arguably one of the most important of these sources and its separation and characterization of the singing voice has a large number of applications in MIR. Most existing query-by-humming systems [4], [5] can only retrieve songs from a database containing music in symbolic format. By performing pitch extraction on the extracted voice signals, it is possible to perform query-by-humming in databases of audio signals. Another potential application is singer identification that is independent of the instrumentation or the "album" effect [6]. Other possible applications include automatic accompaniment, music transcription, and lyrics alignment.

There has been limited work on singing voice separation from monaural recordings. Many existing systems require predominant pitch detection in order to perform separation [7] or rely on prior source models [8]. Other approaches are based on statistical methods such as Independent Component Analysis (ICA) and Non-Negative Matrix Factorization (NMF) [9]. The non-stationarity of the singing voice and music signals as well as their heavy computational requirements are some of the challenges of applying statistical methods to this problem. Another related research area is predominant pitch estimation and melody transcription in polyphonic audio [10] in which only the pitch of the predominant melody or singing voice is estimated.

In contrast, our approach attempts to directly separate the prominent melodic source without first estimating the predominant pitch based on basic perceptually-inspired grouping cues inspired by ideas from Auditory Scene Analysis [11]. A separation of the leading voice is achieved by this approach since the singing voice (if any) is usually the most prominent source of the mixture. A description of an earlier version of the algorithm and some of the experiments described in section III-B appear in [12].

A fundamental characteristic of the human hearing system is the ability to selectively attend to different sound sources in complex mixtures of sounds such as music. The goal of computational auditory scene analysis (CASA) [13] is to create computer systems that can take as input a mixture of sounds and form packages of acoustic evidence such that each package most likely has arisen from a single sound source. Humans use a variety of cues for perceptual grouping in hearing such as similarity, proximity, harmonicity and common fate. For example, sound components that change frequency by the same amount are more likely to be grouped together as belonging to the same sound source (common fate) than if they are changing independently. As another example, two notes played by the same type of instrument are more likely to be grouped together than two notes played on different instruments (similarity). An excellent overview of the current state of the art in CASA is provided in [14].

This work was supported by the National Science and Research Council of Canada (NSERC), the Canada Foundation for Innovation (CFI), the Portuguese Foundation for Science and Technology (FCT), and the Fundação Calouste Gulbenkian (Portugal).

M. Lagrange, J. Murdoch and G. Tzanetakis are with the University of Victoria, Canada (email: {lagrange,jmurdoch, gtzan}@uvic.ca). L. G. Martins is with INESC Porto / FEUP, Portugal (email: lmartins@inescporto.pt).

Many of the computational issues of perceptual grouping for hearing are still unsolved. In particular, considering the several perceptual cues altogether is still an open issue [15], [16]. We propose in this paper to cast this problem into a graph cut formulation using the *normalized cut* criterion. This global criterion for graph partitioning has been proposed for solving similar grouping problems in computer vision [17].

The normalized cut is a representative example of spectral clustering techniques which use an *affinity matrix* to encode topological knowledge about a problem. Spectral clustering approaches have been used in a variety of applications including high performance computing, web mining, biological data, image segmentation and motion tracking.

To the best of our knowledge there are few applications of spectral clustering to audio processing. It has been used for the unsupervised clustering of similar sounding segments of audio [18], [19]. In these approaches, each audio frame is characterized by a feature vector and a self-similarity matrix across frames is constructed and used for clustering. This approach has also been linked to the singular value decomposition of feature matrices to form audio basis vectors [20]. These approaches characterize the overall audio mixture without using spectral clustering to form and track individual sound sources.

Spectral clustering has also been used for blind one-microphone speech separation [21], [22]. Rather than building specific speech models, the authors show how the system can separate mixtures of two speech signals by learning the parameters of affinity matrices based on various harmonic and non-harmonic cues. The entire STFT magnitude spectrum is used as the underlying representation.

Closer to our approach, harmonicity relationships and common fate cues underlie a short-term spectra-based similarity measure presented by Srinivasan [23]. To integrate time constraints, it is alternatively proposed in [24] to cluster previously tracked partials to form auditory "blobs" according to onset cues. Normalized cut clustering is then carried out on these blobs. In contrast, a short-term sinusoidal modeling framework is used in our approach. It results in more accurate and robust similarity relations as well as significantly smaller affinity matrices that are computationally more tractable.

Sinusoidal modeling is a technique for analysis and synthesis whereby sound is modeled as the summation of sine waves parameterized by time-varying amplitudes, frequencies and phases. In the classic McAulay and Quatieri method [25], these time varying quantities are estimated by performing a short-time Fourier transform (STFT) and locating the peaks of the magnitude spectrum. Partial tracking algorithms track the sinusoidal parameters from frame to frame, and determine when new partials begin and existing ones terminate [26]. If the goal is to identify potential sound sources then a separate stage of partial grouping is needed. Typically grouping cues such as common onsets and spectral proximity are used.

In this paper we use the term sound source tracking and formation to refer to these two processes of connecting peaks over time to form partials (tracking) and grouping them to form potential sound sources (formation). They roughly correspond to the sequential and simultaneous aspects of organization described by Bregman [11]. Although frequently implemented as separate stages as in [23], [24], these two organizational principles directly influence one another. For example, if we have knowledge that a set of peaks belong to the same source, then their correspondence with the next frame is easier to find. Similarly, the formation of sound sources is easier if peaks can be tracked perfectly over time. Methods that apply these two stages in a fixed order tend to be brittle as they are sensitive to errors and ambiguity.

To cope with this chicken-and-egg problem, we show how both sound source tracking and formation can be jointly optimized within a unified framework using the normalized cut criterion. We model the problem as a weighted undirected graph, where the nodes of the graph are the peaks of the magnitude spectrum and an edge is formed between each pair of nodes. The edge weight is a function of the similarity between nodes and utilizes various grouping cues such as frequency, amplitude proximity and harmonicity. We also propose a novel harmonicity criterion that we term Harmonically Wrapped Peak Similarity (HWPS), that is described in section II-D. Clustering is performed in the same way for all peaks within a longer "texture window" independently of whether they belong to the same frame or not. The resulting algorithm can be used to separate the singing voice or predominant melody from complex polyphonic mixtures of "real-world" music signals. The algorithm is computationally efficient, causal, and real-time. Another important aspect of our method is that it is data-driven, without requiring a priori models of specific sound sources as many existing approaches to separation do [15].

The remainder of the paper is organized as follows. In the next section, singing voice tracking and formation is formulated as a spectral clustering problem using sinusoidal peaks as the underlying representation. Using this formulation several perceptual grouping criteria such as amplitude proximity, frequency proximity and harmonicity are integrated into a unified framework. Section III describes experimental results demonstrating the potential of the proposed algorithm as a front end for MIR tasks, and conclusions are given in Section IV.

II. Singing Voice Formation and Tracking using the Normalized Cut

A. System Overview

In this section we provide an overview of our proposed method and define the terminology used in the remainder of this article. Figure 1 shows a block diagram of the process; each step of the process is described in more detail in the following subsections. The following terms are important for understanding these descriptions.

- **Frames or analysis windows** are used to estimate sinusoidal peaks from the complex spectrum computed using a Short Time Fourier Transform. For the experiments described in this paper a frame size corresponding to 46 ms and a hop size of 11 ms are used.
- **Peaks** are the output of the sinusoidal modeling stage. For each frame, a variable number of peaks corresponding to

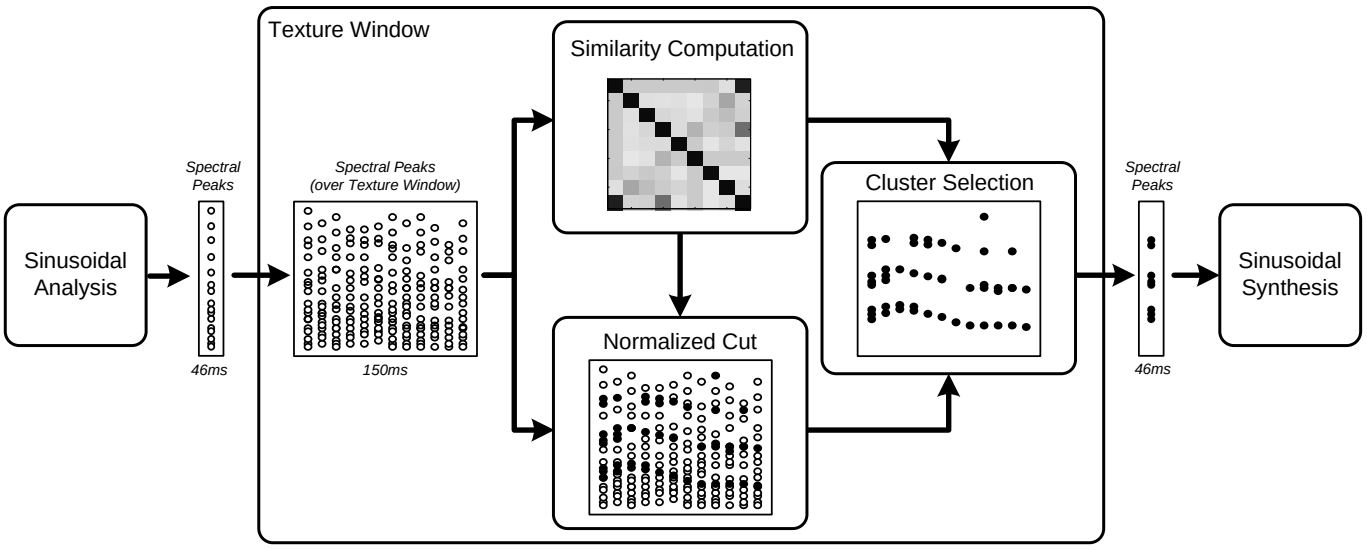

Fig. 1. Block-diagram of the voice segregation chain.

the local maxima of the spectrum are estimated. Each peak is characterized by amplitude, frequency and phase.

- **Texture windows** correspond to an integer number of frames. Clustering of peaks across both frequency and time is performed for each texture window rather than per frame. For the experiments described in this paper a texture window corresponding to 10 frames (≈ 150 ms) is used.
- **Similarity cues** are used to calculate the similarity between sinusoidal peaks belonging to the same texture window. These cues are inspired by perceptual grouping cues [11] such as amplitude and frequency proximity, and harmonicity.
- **The similarity or affinity matrix** is calculated by considering the similarity of every peak to every peak within a texture window. Hence, the similarity matrix represents the similarity between peaks within the same frame (simultaneous integration) and across time (sequential integration) within the "texture window".
- **Clusters** are groups of peaks that are likely to originate from the same sound source. By approximately optimizing the *normalized cut criterion*, the overall peak similarity within a cluster is maximized and the similarity between clusters is minimized. The audio corresponding to any set of peaks (one or more clusters) can be conveniently resynthesized using a bank of sinusoidal oscillators.
- **Source Formation** is the process of approximately reconstructing a particular sound source from a decomposition of the polyphonic mixture.

B. Sinusoidal Modeling

Sinusoidal modeling aims to represent a sound signal as a sum of sinusoids characterized by amplitudes, frequencies, and phases. A common approach is to segment the signal into successive frames of small duration so that the parameters can be considered as constant within the frame. The discrete signal $x^k(n)$ at frame index k is then modeled as follows:

$$x^k(n) \quad = \quad \sum_{l=1}^{L^k} a_l^k \ \cos\left(\frac{2\pi}{F_s} f_l^k \cdot n + \phi_l^k\right) \qquad (1)$$

where F_s is the sampling frequency, ϕ_l^k is the phase at the beginning of the frame of the l-th component of L_k sinusoids, and f_l^k and a_l^k are respectively the frequency and the amplitude. Both are considered as constant within the frame.

For each frame k, a set of sinusoidal parameters $\mathcal{S}^k = \{p_1^k, \cdots, p_{L^k}^k\}$ is estimated. The system parameters of this Short-Term Sinusoidal (STS) model $\mathcal{S}^k$ are the L^k triplets $p_l^k = \{f_l^k, a_l^k, \phi_l^k\}$, often called *peaks*. These parameters can be efficiently estimated by picking some local maxima from a Short-Term Fourier Transform (STFT).

We further improve the precision of these estimates by using phase-based frequency estimators which utilize the relationship between phases of successive frames [27], [28], [29]. Assuming that the frequencies of the pseudo-periodic components of the analysed signal are constant during the time-interval between two successive short-term spectra, with a hop size of H samples, the frequency can be estimated from the phase difference:

$$\hat{\omega} = \frac{1}{2\pi H}\Delta\phi \qquad (2)$$

The smaller the hop size the more accurate is this assumption. We consider two successive short-term spectra separated by one sample. The frequency is estimated directly from the phase difference $\Delta\phi$ ensuring that the phase is unwrapped so that the difference is never negative. The resulting estimator, known as the difference estimator, is:

$$f_l^k = \frac{F_s}{2\pi}\ (\angle S[m_l, n_k + 1] - \angle S[m_l, n_k])_{\text{unwrap}} \qquad (3)$$

where $\angle S$ denotes the phase of the complex spectrum S, m_l is the frequency bin index of the peak p_l^k within the spectrum, and n_k is the index of the first sample of analysis frame k. The frame index k is omitted in the remainder of this subsection.

During the analysis of natural sounds, presence of several frequency components in the same spectral bin or noise may lead to incoherent estimates. If the frequency f_l of a local maximum located at bin m_l is closer to the frequency of another bin, the local maximum should have been located at this bin. Therefore, a local maximum with an estimated frequency that does not satisfy the following condition is discarded: $|N/F_s \cdot f_l - m_l| \leq 0.5$.

Since that the power spectrum of an ideal sinusoid signal has the shape of the power spectrum of the analysis window, centered around the sinusoid frequency, the increase of frequency precision can be used to estimate more precisely the amplitude:

$$a_l = 2\,\frac{|S[m_l]|}{|W_H(f_l - m_l\,F_s/N)|} \tag{4}$$

where $|S[m_l]|$ is the magnitude of the complex spectrum, $W_H(f)$ is the spectrum of the Hann window used for the STFT computation, f being the frequency in Hz. These more precise estimates of frequency and amplitude parameters are important for calculating more accurate similarity relations between peaks. Other frequency estimation methods can also be used, such as parabolic interpolation [30] or subspace methods [31]. The importance of more precise frequency estimation for our method compared to the basic approach of directly converting the FFT frequency bin number to frequency is explored in some of the experiments in Section III.

Sinusoidal modeling is particularly suited for sustained sounds with a definite pitch and harmonic structure such as the vowels of a singing voice. Consonants can still be represented but require a large number of sinusoidal components to be represented accurately [25]. Vowels in singing voices tend to last much longer than in speech therefore in practice a sinusoidal model can capture most of the singing voice information that is useful for MIR applications such as the melodic line and the timbral characteristics.

C. Grouping Criteria

In order to simultaneously optimize partial tracking and source formation we construct a graph over each "texture" window. Unlike approaches based on local information [25], we utilize the global normalized cut criterion to partition the graph over the entire "texture" window. Edges are formed both for peaks within a frame and peaks across frames. Each partition is a set of peaks that are grouped together such that the similarity within the partition is maximized and the similarity between different partitions is minimized. If L^k is the number of peaks in frame k, then the number of peaks in a "texture" window is T resulting in a similarity matrix of size $T * T$:

$$T = \sum_k L^k \tag{5}$$

The maximum L^k used in our experiments is set to 20. By picking the highest amplitude peaks of the spectrum, we usually achieve fair resynthesis quality using a small number of sinusoids per frame with a significant savings in computation time. For example if the entire STFT spectrum is used as in Bach [21] the similarity matrix for 10 analysis frames of 512 samples would have a size of 5120 * 5120 whereas in our case would be have a maximum possible size of 200 * 200.

TABLE I
HARMONIC SOURCES USED FOR FIGURES 2, 3, 4, 5

	A0	A1	A2	A3	A4,B3	B0	B1	B2	B4
f	440	880	1320	1760	2200	550	1100	1650	2750
a	.8	.8	.6	.4	.4	1	.8	.6	.4

In a first approach, the edge weight connecting two peaks p_l^k and p_m^{k+n} depends on both frequency W_f and amplitude W_a proximity (k is the frame index and l,m are peak indices, $n \in \{0 \ldots N-1\}$ is the frame offset between the peaks, and N is the "texture" window size; $n = 0$ is used for peaks of the same frame):

$$W_{fa}(p_l^k, p_m^{k+n}) = W_f(p_l^k, p_m^{k+n}) * W_a(p_l^k, p_m^{k+n}) \tag{6}$$

We use radial basis functions (RBFs) to model the frequency and amplitude similarities:

$$W_{fa}(p_l^k, p_m^{k+n}) = e^{-\left(\frac{f_l^k - f_m^{k+n}}{\sigma_f}\right)^2} * e^{-\left(\frac{a_l^k - a_m^{k+n}}{\sigma_a}\right)^2} \tag{7}$$

The standard deviations of frequencies and amplitudes are calculated separately for each texture window. For these two similarities the amplitudes are measured in Decibels (dB) and the frequencies are measured in Barks (approximately linear below 500 Hz and logarithmic above). Amplitude and frequency cues are not enough for multiple overlapping harmonic sound sources. In the following subsection we describe a harmonic similarity measure between peaks that works well for these cases.

D. Harmonically Wrapped Peak Similarity

A wide variety of sounds produced by humans are harmonic, from singing voice and speech vowels, to musical sounds. As a result the harmonicity cue has been widely studied. As explained by de Cheveigné in [14] for the case of multiple fundamental frequency estimation, most approaches use an iterative method whereby the fundamental frequency of the dominant source is identified and then used to remove the corresponding source from the mixture. Few studies have focused on the identification of harmonic relations between peaks without any prior fundamental frequency estimation.

The goal is to define a similarity measure between two frequency components (peaks) that is high for harmonically related peaks and low for peaks that are not harmonically related. Most existing approaches [32], [23], [33], [34] use the mathematical properties of the harmonically related frequencies to build such a similarity measure for a single frame. For example, Virtanen [32] considers whether the ratio of the frequencies of the components is a ratio of small positive integers, while Martins [34] selects peaks that are equally far apart in frequency to form harmonic clusters.

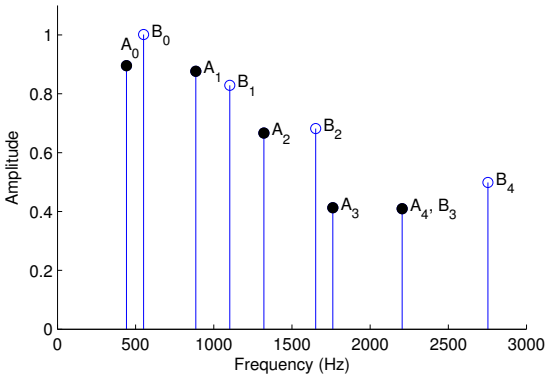

Fig. 2. Two sets of harmonically related peaks (same data as Table I). Used for Figures 3,4,5.

There are several issues concerning these approaches, both from the technical and perceptual points of view. First, these type of measures can not be safely considered for peaks belonging to different frames, which is a strong handicap for our application. The reason of this restriction is that the fundamental frequency of the source can change across frames. Secondly, these mathematical conditions are not sufficient to determine whether two peaks are part of an harmonic source. From a perceptual point of view, two peaks are close on the "harmonic" axis if these peaks belong to a perceptible compound of harmonically-related peaks in the spectrum. This fact perhaps explains why most separation algorithms first attempt to identify the pitch of the sounds within the mixture by considering the spectral information globally, and then assign frequency components to each estimated pitch. In contrast, our proposed similarity measure works reasonably well without estimating the underlying pitch.

To address these problems, we introduce a new similarity measure that we term Harmonically Wrapped Peak Similarity (HWPS). The main goal of the HWPS measure is to take advantage of the flexibility of an harmonically-related similarity between peaks that not only considers each peak in isolation but also the entire spectral information associated with the remaining peaks. This measure can be used both for peaks within the same frame and among peaks of different frames.

The basic mechanism behind the HWPS measure is to assign each peak a spectral pattern. The pattern captures information about the spectrum in relation to the specific peak. The degree of matching between two spectral patterns is used as a similarity measure between the two peaks thus utilizing more spectral information than just the amplitude and frequency of the two peaks. As the spectral pattern of a peak might shift when changing frames and contains peaks belonging to multiple harmonic sources we use a harmonically wrapped frequency space to align the two spectral patterns corresponding to the peaks. The goal is that the similarity between peaks belonging to the "same" harmonic complex is higher than the similarity of peaks belonging to different harmonic complexes. The three steps of this process are described below in more detail:

Shifted Spectral Pattern: Our approach relies on a description of the spectral content using estimates of the frequency and amplitude of local maxima of the power spectrum, i.e. the peaks. We therefore propose to assign to each peak, p_l^k (l is the peak index, and k is the frame index), a given spectral pattern, $\tilde{F}_l^k$, based on the set of frequencies (in Hz), $F_l^k = \{f_i^k\}$, shifted within the frame k as follows:

$$\tilde{F}_l^k = \{\tilde{f}_i^k | \tilde{f}_i^k = f_i^k - f_l^k, \forall i \in [1, L_k]\} \tag{8}$$

where L_k is the highest peak index of frame k.

The spectral pattern is essentially a shift of the set of peak frequencies such that the frequency of the peak corresponding to the pattern maps to 0 (when i is equal to l). One can easily see that two peaks of different frames modeling the same partial will have roughly similar spectral patterns under the assumption that the spectral parameters evolve slowly with time. This spectral pattern forms a peak-specific view of the spectral content which is used to calculate a pitch invariant representation using a wrapped frequency space as described in the following subsection. The top graphs of Figures 3 and 4 show overlaid peak-specific spectral patterns for two pairs of peaks from the harmonic mixture of Figure 2.

Wrapped Frequency Space: To estimate whether two peaks p_l^k and p_m^{k+n} belong to the same harmonic source, we propose to measure the correlation between the two spectral patterns corresponding to the peaks. To achieve this, we would like to transform the peak-specific spectral patterns in such a way that when the peaks under consideration belong to the same harmonic complex the correlation is higher than when they belong to different harmonic sources. In order to achieve this the following operations are performed: the energy distribution of an harmonic source along the frequency axis can be seen as a cyclic unfolding with periodicity equal to the fundamental frequency of the source. To concentrate these energies as much as possible before correlating them, we propose to wrap the frequencies of each spectral pattern as follows:

$$\hat{f}_i^k = \mathrm{mod}\left(\frac{\tilde{f}_i^k}{h}, 1\right) \tag{9}$$

where h is the wrapping frequency function and mod is the real modulo function. This wrapping operation would be perfect with the prior knowledge of the fundamental frequency. With this knowledge we can parametrize the wrapping operation with:

$$h = \min(\mathrm{f}_{0l}^k, \mathrm{f}_{0m}^{k+n}) \tag{10}$$

where f_{0l}^k is the fundamental frequency of the source of the peak p_l^k. Without such prior, we consider a conservative approach which tends to over estimate the fundamental frequency with:

$$h' = \min(f_l^k, f_m^{k+n}) \tag{11}$$

Notice that the value of the wrapping frequency function h is the same for both patterns corresponding to the peaks under consideration. Therefore the resulting wrapped frequency spectra will be more similar if the peaks belong to the same harmonic source. The resulting wrapped frequency spectra are pitch invariant and can be seen in the middle plot of Figures 3 and 4.

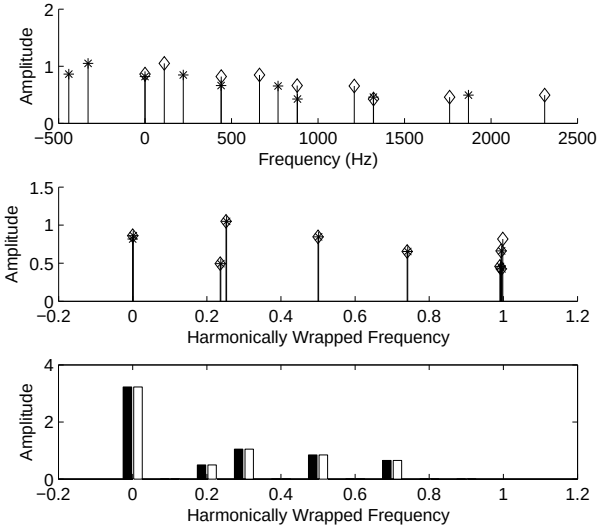

Fig. 3. HWPS calculation for peaks A_0 $\diamondsuit$ and A_1 $*$, from Figure 2. From top to bottom: Shifted Spectral Pattern, Harmonically-Wrapped Frequency and Histogram of Harmonically-Wrapped Frequency. Notice the high correlation between the two histograms at the bottom of the figure.

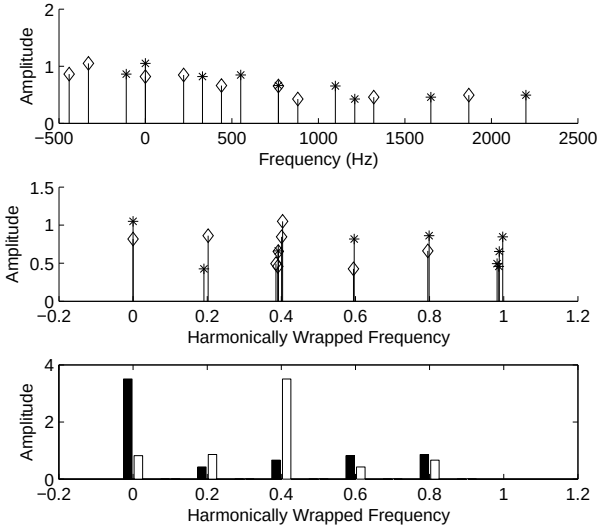

Fig. 4. HWPS calculation for peaks A_1 $\diamondsuit$ and B_1 $*$, from Figure 2. From top to bottom: Shifted Spectral Pattern, Harmonically-Wrapped Frequency and Histogram of Harmonically-Wrapped Frequency. Notice the lack of correlation between the two histograms at the bottom of the figure.

Discrete Cosine Similarity: The last step is now to correlate the two harmonically wrapped spectral patterns ($\hat{F}_l^k$ and $\hat{F}_m^{k+n}$) to obtain the HWPS measure between the two corresponding peaks. This correlation can be done using an algorithmic approach as proposed in [35], but this was found not to be reliable or robust in practice. Alternatively, we propose to discretize each harmonically wrapped spectral pattern into an amplitude weighted histogram, H_l^k, corresponding to each spectral pattern $\hat{F}_l^k$. The contribution of each peak to the histogram is equal to its amplitude and the range between 0 and 1 of the Harmonically-Wrapped Frequency is divided into 20 equal-size bins. In addition, the harmonically wrapped spectral patterns are also folded into an octave to form a pitch-invariant chroma profile. For example, in Figure 3, the energy of the spectral pattern in wrapped frequency 1 (all integer multiples of the wrapping frequency) is mapped to histogram bin 0.

The HWPS similarity between the peaks p_l^k and p_m^{k+n} is then defined based on the cosine distance between the two corresponding discretized histograms as follows:

$$W_h(p_l^k, p_m^{k+n}) = e^{\left(\frac{c(H_l^k, H_m^{k+n})}{\sqrt{c(H_l^k, H_l^k)\cdot c(H_m^{k+n}, H_m^{k+n})}}\right)^2} \quad (12)$$

where

$$c(H_a^b, H_c^d) = \sum_i H_a^b(i) * H_c^d(i). \quad (13)$$

One may notice that due to the wrapping operation of Equation 9, the size of the histograms can be relatively small (around 20), thus saving computational time.

To illustrate this, let us consider the case of the mixture of two pitched sound sources, A and B, each composed of 4 harmonics with fundamental frequencies of 440 Hz and 550 Hz respectively, as presented in Table I and Figure 2. For the experiments, random frequency deviations of a maximum of 5 Hz are added to test the resilience of the algorithm to frequency estimation errors. If we consider two peaks of the same source A_0 and A_1, the quantized version of the harmonically-wrapped sets of peaks are highly correlated, as can be seen in the bottom of Figure 3. On the other hand, if we consider two peaks of different sources, A_1 and B_0, the correlation between the two discretized histograms is low (see Figure 4). The correlation between two histograms of harmonically related peaks still works (although to a lesser extent) if instead of using the true fundamental f_0 as the wrapping frequency we use any harmonic of it.

Figure 5 (left) shows a HWPS similarity matrix computed among the peaks of two overlapping harmonic sounds within a frame (also shown in Figure 2) with perfect knowledge of the fundamental frequency for each peak respectively. As can be seen clearly from the figure, the similarity is high for pairs of peaks belonging to the same source and low for pairs belonging to different sources. Figure 5 (right) shows the HWPS similarity matrix computed among the peaks of the same two overlapping harmonic sounds within a frame using the conservative approach to estimate the wrapping frequency (basically considering the lower peak as the "wrapping" frequency). As can be seen from the figure, although the similarity matrix on the right is not as clearly defined as the one on the left, it still clearly shows higher values for pairs of peaks belonging to the same sound source.

E. Spectral Clustering and the Normalized Cut Criterion

The normalized cuts algorithm, presented in [17], aims to partition an arbitrary set of data points into n clusters. The data set is modeled as a complete weighted undirected graph

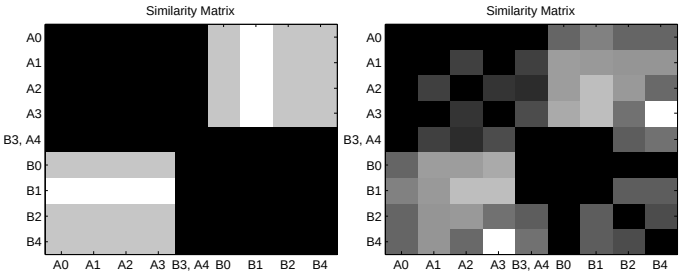

Fig. 5. Harmonically-Wrapped Peak Similarity (HWPS) matrix for two harmonic sources using the correct f_0 estimates (left), and using the conservative estimate of wrapping frequency (likely a harmonic of the "true" f_0) (right). High similarity values are mapped to black and low similarity values to white.

G = (V, E), the nodes V representing the data points and each edge E weight, $w(i, j)$, representing the relative similarity between the two nodes i and j. The graph is represented internally by an affinity matrix, W, that specifies all edge weights. The partitioning is achieved by recursively dividing one of the connected components of the graph into two until n complete components exist. The formulation of the $Ncut$ measure addresses the bias towards partitioning out small sets of isolated nodes in a graph which is inherent in the simpler minimal cut disassociation measure (cut) between two graph partitions A, B:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{14}$$

The criterion that is minimized in order to establish the optimal partitioning at any given level is the normalized cut disassociation measure ($Ncut$):

$$Ncut(A, B) = \frac{cut(A, B)}{\text{asso}(A, V)} + \frac{\text{cut}(A, B)}{\text{asso}(B, V)} \tag{15}$$

where $\text{asso}(X, V) = \sum_{u \in X, t \in V} w(u, t)$ is the total of the weights from nodes in cluster X to all nodes in the graph. An analogous measure of the association within clusters is the following (Nasso):

$$\text{Nasso}(A, B) = \frac{\text{asso}(A, A)}{\text{asso}(A, V)} + \frac{\text{asso}(B, B)}{\text{asso}(B, V)} \tag{16}$$

where $\text{asso}(X, X)$ is the total weight of edges connecting nodes within cluster X. We note the following relationship between $Ncut$ and Nasso:

$$\text{Ncut}(A, B) = 2 - \text{Nasso}(A, B) \tag{17}$$

Hence, the attempt to minimize the disassociation between clusters is equivalent to maximizing the association within the clusters. The hierarchical clustering of the data set via the minimization of the Ncut measure, or the equivalent maximization of the Nasso measure may be formulated as the solution to an eigensystem. In particular, [17] show that the problem is equivalent to searching for an indicator vector $\mathbf{y}$ such that $y_i \epsilon \{1, b\}$ depending on which of the two sub-partitions node i is assigned and b is a function of the sum of total connections of the nodes. By relaxing $\mathbf{y}$ to take on real values, we can find a partition that minimizes the Ncut criterion by solving the generalized eigenvalue system:

$$(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda \mathbf{D}\mathbf{y} \tag{18}$$

where $\mathbf{D}$ is a $N \times N$ diagonal matrix with each element of the diagonal $d(i)$ being the total connection from node i to all other nodes $d(i) = \sum_j w(i, j)$, and $\mathbf{W}$ is a $N \times N$ symmetrical matrix containing the edge weights. As described in Shi and Malik [17] the second smallest eigenvector of the generalized eigensystem of equation (18) is the real valued solution to the *Normalized Cut* minimization. For discretizing the eigenvectors to produce indicator vectors we search among l evenly spaced splitting points within the eigenvector for the one that produces a partition with the best (i.e. smallest) $\text{Ncut}(A, B)$ value. The graph is recursively sub-divided as described above until n clusters of peaks have been extracted.

One of the advantages of the normalized cut criterion for clustering over clustering algorithms such as K-means or mixtures of Gaussians estimated by the EM algorithm is that there is no assumption of convex shapes in the feature representation. Furthermore, the divisive nature of the clustering does not require a priori knowledge of the number of output clusters. Finally, compared to point based clustering algorithms such as K-means, the use of an affinity matrix as the underlying representation enables expression of similarities that can not be computed as a distance function of independently calculated feature vectors. The Harmonically Wrapped Peak Similarity measure proposed in this paper and described in section II-D, is an example of such as a similarity measure.

F. Cluster Selection

Among the several clusters, C_i, identified using the normalized cut clustering for each texture window, we want to select the cluster that most likely contains the voice signal. The proposed approach is straightforward and does not rely on any prior knowledge of the voice characteristics. This criterion is therefore more general and may apply to the selection of any predominant harmonic source.

A cluster of peaks corresponding to a predominant harmonic source should be dense in the feature space in which we compute similarities. The reason is that peaks belonging to a prominent harmonic audio source have more precise parameter estimates and therefore comply better to the implicit model expressed by the various similarity functions. The peaks of a prominent source will therefore tend to be more similar (mostly in terms of harmonicity) to peaks belonging to the same source than to other sources. Thus, the intra-cluster similarities should be high for this particular cluster. Let us consider a cluster of peaks P_c of cardinality $\#P_c$ defined as the set of peaks whose Ncut label is c:

$$P_c = \{p_m^k | \text{label}(p_m^k) = c\} \tag{19}$$

We then consider the density criterion:

$$d(P_c) = \frac{1}{{\#P_c}^2} \sum_{p_l^k \in P_c} \sum_{p_m^j \in P_c} W_{fah}(p_l^k, p_m^j) \tag{20}$$

TABLE II
SDR VALUES FOR OLD+NEW EXPERIMENTS

	XN	XS	VN	VS	CN	CS
A+F	12.87	9.33	10.11	7.67	2.94	1.52
A+F+H	13.05	9.13	11.54	7.69	3.01	2.09

TABLE III
SDR VALUES USING "TEXTURE" WINDOWS

	XN	XS	VN	VS	CN	CS
A+F	9.79	3.09	3.29	6.50	3.01	3.01
A+F+H	7.33	5.03	4.73	5.35	3.08	3.07

where k, j are the frame indices within the texture window and l, m are respectively the peak indices within the frames k and j. The function W_{fah} refers to the overall similarity weight that takes into account frequency, amplitude, and harmonicity. It is the product of the corresponding weights:

$$W_{fah}(p_l, p_m) = W_f(p_l, p_m) * W_a(p_l, p_m) * W_h(p_l, p_m) \quad (21)$$

For the experiments described in section III we computed 3 clusters for each texture window and selected the two clusters with the highest density as the ones corresponding to the voice signal. The peaks corresponding to the selected clusters are used to resynthesize the extracted voice signal using a bank of sinusoidal oscillators.

III. EXPERIMENTAL EVALUATION

The main goal of the experiments described in the following subsections is to demonstrate the potential of the proposed method. The algorithm is efficient and requires no training or prior knowledge. The source code of the algorithm is available as part of *Marsyas*, a cross-platform open source software framework for audio analysis and synthesis[1]. We hope that making the algorithm publicly available will encourage other researchers to use it and experiment with it. We have also made available on a website[2] more information about the method (such as MATLAB code for the HWPS calculation), as well as the audio datasets described in this section.

A. Corpus description and experimental setup

Three datasets were used for the experiments. The first dataset was used for tuning the algorithm and the evaluation of the HWPS similarity cue described in subsection III-B. It consists of synthetically-created mixtures of isolated instrument sounds, voice, harmonic sweeps and noise. Each clip is approximately 1 second long.

The second dataset consists of 10 polyphonic music signals for which we have the original vocal and music accompaniment tracks before mixing as well as the final mix. Although relatively small, this dataset is diverse and covers different styles of singing and music background. It contains the following types of music: rock (6), celtic (3), hiphop (1). In addition, we also evaluate melody extraction on the corpus used for the series of MIREX audio melody extraction evaluation exchanges; it consists of 23 clips of various styles including instrumental and MIDI tracks, for which case we estimate the dominant melodic voice.

[1] http://marsyas.sourceforge.net
[2] http://opihi.cs.uvic.ca/NormCutAudio

B. Evaluation of the HWPS cue

The HWPS is a novel criterion for computing similarity between sinusoidal peaks that are potentially harmonically related. It is critical for separating harmonic sounds which are particularly important for musical signals. In this subsection we describe experiments showing the improvement in separation performance achieved by using the HWPS similarity cue compared to two existing cues proposed in Srinivasan [23] and Virtanen [32].

Existing Cues: Srinivasan consider an harmonicity map that can be precomputed to estimate the harmonic similarity between two spectral bins. Considering two bin indexes i and j, the map is computed as follows:

$$\text{hmap}(i, j) = 1 \text{ if } \text{mod}(i, j) = 0 \text{ or } \text{mod}(j, i) = 0 \quad (22)$$

This map is next smoothed to allow increasing level of inharmonicity using a Gaussian function, normalized so that the sum of its elements is unity. The standard deviation of the Gaussian function is set to be 10% of its center frequency, see [23] for further details. The similarity between peaks p_l and p_m is then:

$$W_s(p_l, p_m) = \text{shmap}(M_l, M_m) \quad (23)$$

where shmap is the smoothed and normalized version of hmap, and M_l corresponding to the bin index of peak p_l. The frames indexes are omitted for clarity sake.

According to Virtanen [32], if two peaks p_l and p_m are harmonically related, the ratio of their frequencies f_l and f_m is a ratio of two small positive integers a and b (which correspond to the harmonic rank of each peak, respectively). By assuming that the fundamental frequency cannot be below the minimum frequency found by the sinusoidal modeling front-end (i.e. $f_{\text{min}} = 50$ Hz), it is possible to obtain an upper limit for a and b, respectively $a < \left\lfloor \frac{f_l}{f_{\text{min}}} \right\rfloor$ and $b < \left\lfloor \frac{f_m}{f_{\text{min}}} \right\rfloor$. A harmonic distance measure can be defined:

$$W_v(p_l, p_m) = 1 - \text{min}_{a,b} \left| \log \left(\frac{f_l / f_m}{a / b} \right) \right| \quad (24)$$

by considering all the ratios for possible a and b and choosing the closest to the ratio of the frequencies.

Evaluation criteria: To compare our proposed cue with the existing ones, we use the signal-to-distortion ratio (SDR) as a simple measure of the distortion caused by the separation algorithm [36]. It is defined in decibels as:

$$SDR[dB] = 10 \log_{10} \frac{\sum_t s(t)^2}{\sum_t [\hat{s}(t) - s(t)]^2} \quad (25)$$

where s(t) is the reference signal with the original separated source and $\hat{s}(t)$ is the extracted source. The main use of the SDR measure in this section is to evaluate the relative improvement in separation performance achieved by the use of the

TABLE IV
SDR MEASUREMENTS OF POLYPHONIC MUSIC SEPARATION USING DIFFERENT SIMILARITY MEASURES

Track Title	AF	HS	HV	rHWPS	HWPS
bentOutOfShape	2.66	1.19	1.17	5.65	8.03
intoTheUnknown	0.65	3.24	0.81	3.29	4.05
isThisIt	1.94	2.71	2.07	2.18	2.65
landingGear	1.29	5.29	0.81	4.40	6.37
schizosonic	0.57	3.11	0.59	2.86	3.96
smashed	0.22	1.15	0.29	1.17	1.54
chavalierBran	4.25	7.21	1.6	4.02	6.83
laFee	7.7	6.62	2.48	4.88	6.93
lePub	0.23	0.48	0.23	0.38	0.47
rockOn	0.96	1.78	0.79	1.60	1.74

HWPS cue. The SDR is only an approximate measure of the perceptual quality of the separated signals. We also encourage the readers to listen to the examples on the provided webpage. No post-processing of the extracted signals was performed in order to provide better insight about the algorithm and its limitations. For example a dominant cluster is always selected independently of the presence of a singing voice.

For the first set of experiments we utilize an experimental setup inspired by the "old+new" heuristic described by Bregman [11]. Similar experiments were described in [12]. Each sample is created by mixing two sound sources in the following way: for the first part of the sound only the "old" sound source is played followed by the addition of the "new" sound source (old+new) in the second part of the sample. *Normalized cut* clustering is performed over the entire duration of the clip. The clusters that contain peaks in the initial "old"-only part are selected as the ones forming the separated source. The remaining peaks are considered to be part of the "new" sound source. The SDR measures the distortion/interference caused by this "new" sound source to the separation algorithm.

Table II compares different mixtures of isolated sounds separated using only frequency and amplitude similarities, and also separated with the additional use of the HWPS similarity. The following conventions are used in Table II: X is saxophone, N is noise, V is violin, S is harmonic sweep, and C is voice. A, F, H correspond to using amplitude, frequency and HWPS similarities, respectively. As can be seen from the table, in almost all cases the use of the HWPS improves the SDR measure of separation performance.

For the second set of experiments shown in Table III the "old+new" mixtures are separated directly using the approach described in subsection II-F to select the dominant sound source. Unlike the previous experiments, the spectral clustering is performed separately for each "texture" window and the highest density cluster is selected as the separated voice. This is a more realistic scenario as no knowledge of the individual sound sources is utilized. As expected the SDR values are lower but again the use of the HWPS improves separation performance in most cases.

The third set of experiments, shown in Table IV, illustrate the improvement in separation performance using the HWPS cue for the case of singing voice extraction from monaural polyphonic audio recordings. As a reference signal for computing the SDR we use a sinusoidal representation of the original voice-only track using 20 sinusoidal peaks per analysis frame. This way the SDR comparison is between similar representations and therefore more meaningful. This representation of the reference signal is perceptually very similar to the original voice-only signal and captures the most important information about the singing voice, such as the identity of the singer, pitch, vibrato, etc. The first column of Table IV shows the performance of our approach using only the amplitude and frequency similarities. The other columns show the performance of using the three different harmonicity similarities in addition to amplitude and frequency. All the configurations utilize the same parameters for the Normalized Cut algorithm and the only thing that changes is the definition of the similarity function. As can be seen, in most cases the *HWPS* similarity provides better results than the Virtanen similarity (*HV*) and the Srinivasan similarity (*HS*) and behave similarly otherwise. Finally the last two column show the importance of precise frequency estimation (*HWPS*) described in section II-B compared to rough frequency estimation directly from FFT bins (*rHWPS*). A similar drop in performance between rough and precise estimation was also observed for *HV* and *HS* but not included in the table.

C. Melodic Pitch Extraction

The melodic line is a critical piece of information for describing music and is very influential in the identity of a musical piece. A common approach to automatic melody extraction is to attempt multiple pitch extraction on the polyphonic mixture and select the predominant pitch candidate as the pitch of the signing voice [7], [37]. The detected pitch can then be used to inform source separation algorithms. In our method the singing voice is first separated and the melodic pitch is subsequently extracted directly from the separated audio.

For each song in our dataset of 10 songs (for which the original voice-only tracks are available) the pitch contours were calculated for three configurations: the original clean vocal signal, the polyphonic recording with both music and vocals (VM), and the vocal signal separated by our algorithm ($VSep$). Two pitch extraction algorithms were utilized: a time domain autocorrelation monophonic pitch extraction algorithm implemented in Praat [38], and a recent multipitch estimation algorithm developed by Klapuri [37]. Both approaches were configured to estimate fundamental frequencies in the range $[40, 2200]$ Hz, using a hop size of 11ms and an analysis window with a length of about 46ms.

The pitch contours estimated using Praat from the polyphonic recordings with both music and vocals will be referred in the text, figures, and tables as VM_{praat}, while the ones extracted using Klapuri's algorithm will be referred as VM_{klap}. Similarly, for the separated vocal signals we use $VSep_{praat}$. For ground truth we extract a reference pitch contour using Praat from the original voice-only track of each song. We confirmed that this ground truth is correct by listening to the generated contours.

For the purpose of this specific evaluation we only consider the singing segments of each song, identified as the segments

TABLE V
NORMALIZED PITCH ERRORS AND GROSS ERRORS ACROSS CORPUS

	NE	NE_{chr}	$GE(\%)$	$GE-8^{ve}(\%)$
VM_{praat}	8.62	0.51	82.44	66.00
$VSep_{praat}$	3.89	0.35	64.45	55.23
VM_{klap}	0.55	0.26	55.70	48.68

in the ground truth pitch contours that present non-zero frequency values. For each pitch contour we compute at each frame the normalized pitch error as follows:

$$NE[k] = \left| \log_2 \frac{f[k]}{f_{\text{ref}}[k]} \right| \tag{26}$$

where $f_{ref}[k]$ corresponds to the frequency values of the ground truth pitch contour, measured in Hertz, and k is the frame index. $f[k]$ is either related to the frequency value of the pitch contour extracted from the mixed signal (i.e. VM_{praat} or VM_{klap}), or to the pitch contour extracted using the vocal track separated by our algorithm (i.e. $VSep_{praat}$). The normalized error NE is zero when the pitch estimation is correct, and an integer number for octave errors (i.e. when $f[k] = 2^n \times f_{ref}[k], n \in \mathbb{N}$). Since we are only considering the singing segments of the signals, both $f[k]$ and $f_{ref}[k]$ will never be zero.

Given that this evaluation is related to (musical) pitch estimation, we also defined a chroma-based error measure NE_{chr} derived from NE, where errors are folded into a single octave as follows:

$$NE_{chr}[k] = \begin{cases} 0 & \text{if} \quad NE[k] = 0 \\ 1 & \text{if} \quad NE[k] \neq 0 \ \wedge \ \text{mod}(NE[k], 1) = 0 \\ \text{mod}(NE[k], 1) & \text{otherwise} \end{cases} \tag{27}$$

where mod() is the real modulo function. This allows bringing to evidence the chromatic distribution of the errors, wrapping all pitch inaccuracies into the interval $[0, 1]$, where 0 corresponds to no pitch estimation error and 1 accumulates all the octave-related errors.

Table V shows the normalized pitch errors across our dataset of 10 songs for the different evaluation scenarios. It also presents the gross error (GE) for each case, defined as the sum of all errors bigger than half a semitone (i.e. for $NE > \frac{1}{24}$). This tolerance allows accepting estimated frequencies inside a one semitone interval centered around the true pitch as correct estimates. Also presented is the gross error excluding all octave errors ($GE - 8^{ve}$). Octave ambiguities can be accepted as having smaller impact than other errors for many musical applications.

From the results obtained from the VM_{praat} evaluation, a GE in excess of 82% confirms the expected inability of monophonic pitch detectors such as the Praat algorithm to accurately estimate the most prominent pitch on polyphonic music recordings. However, if using the same exact pitch estimation technique on the voice signal separated by our system (i.e. $VSep_{praat}$), the results demonstrate a clear improvement, reducing the gross error rate GE to about 64%. Although the proposed scheme do not compare favourably to the state-of-the-art multipitch algorithm by Klapuri (GE of 55.7%), it shows the ability of our method to simplify the acoustic scene by focusing on the dominant harmonic source.

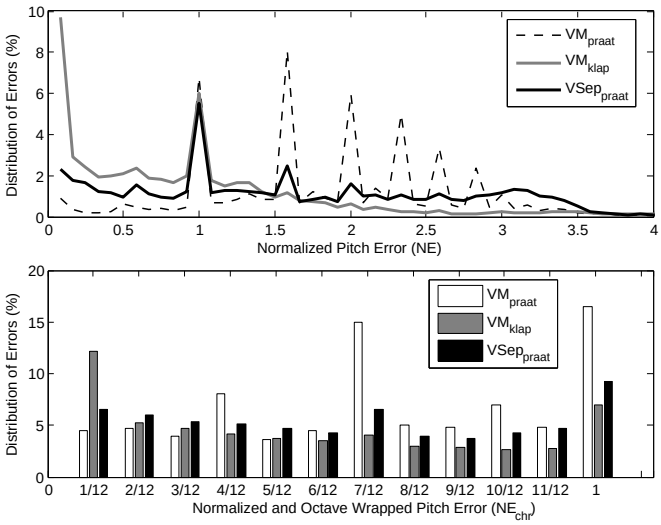

Fig. 6. Distribution of errors as percentages of total pitch estimates across corpus. The top plot presents a segment of the normalized error distribution, NE (no significant error values exist outside the plotted ranges), while the bottom plot depicts the corresponding octave wrapped error distribution, NE_{chr}. The pitch errors from using Praat on the separated voice signals are represented in black colour, while its use on mixed signals is represented by the slashed line and white bars. The multiple pitch approach of Klapuri is represented in gray.

It is also interesting to look at the distribution of errors. Figure 6 shows the distribution of the normalized and octave wrapped errors as percentages over the total number of estimated pitch frames (the upper plot presents the significant section of the NE distribution while the lower plot shows the NE_{chr} distribution). All three evaluations presented in the plots show a similar tendency to output one-octave ambiguities (i.e. about 6% for $NE = 1$). VM_{praat} presents several additional high-valued error peaks caused by incorrect pitches estimated due to the presence of multiple overlapping notes from the musical background. These errors are significantly reduced in the case of the pitch estimation on the separated signal using our method. When compared to VM_{klap} most of the pitch estimation errors from $VSep_{praat}$ result from octave and perfect-fifth (i.e $NE_{chr} = 7/12$) ambiguities.

We also conducted similar experiments using the corpus used for the MIREX automatic melody extraction evaluation exchange[3]. In this case we had no access to the original melody-only tracks, but ground truth pitch contours were provided for the evaluation. The MIREX examples include some synthesized MIDI pieces, which are simpler to separate as they do not include reverberation and other artifacts found in realworld signals. Most of the examples also have a more pronounced vocal line or dominant melody than the corpus used for the previous experiments, and therefore most of the results were better. Table VI shows the normalized pitch errors and gross errors for the MIREX corpus. The distribution of the normalized errors are depicted in Figure 7.

[3] http://www.music-ir.org/mirex2005/index.php/Main_Page

TABLE VI
NORMALIZED PITCH ERRORS AND GROSS ERRORS FOR MIREX DATASET

	NE	NE_{chr}	$GE(\%)$	$GE-8^{ve}(\%)$
VM_{praat}	3.29	0.48	76.02	55.87
$VSep_{praat}$	1.34	0.36	54.12	34.97
VM_{klap}	0.34	0.15	34.27	29.77

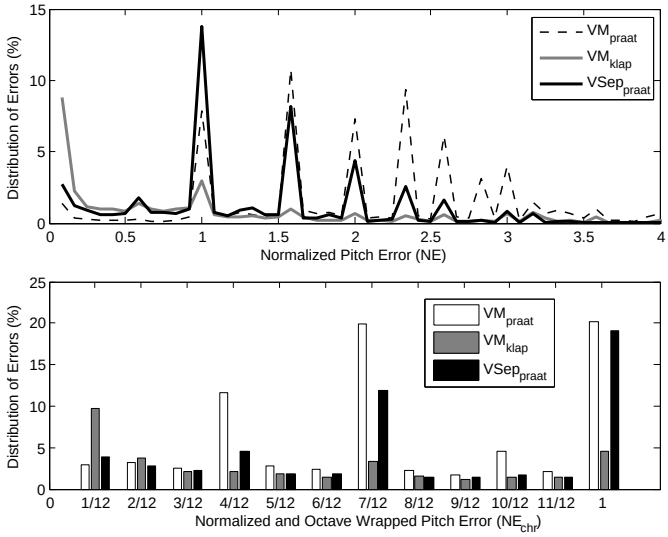

Fig. 7. Distribution of errors over the MIREX audio melody extraction dataset. The top plot presents a segment of the normalized error distribution, NE (no significant error values exist outside the plotted range), while the bottom plot depicts the corresponding octave wrapped error distribution, NE_{chr}. The pitch errors from using Praat on the separated voice signals are represented in black colour, while its use on mixed signals is represented by the slashed line and white bars. The multiple pitch approach of Klapuri is represented in gray.

D. Voicing Detection

Voicing detection refers to the process of identifying whether a given time frame contains a "melody" pitch or not. The goal of the experiments described in this subsection was to determine whether the proposed voice separation algorithm can be used to improve voicing detection accuracy in monaural polyphonic recordings. The dataset of the 10 polyphonic music pieces for which we have the original separate vocal track was used for the experiments. The voiced/unvoiced decisions extracted using Praat [38] from the original vocal track were used as the ground truth. A supervised learning approach was used to train voiced/unvoiced classifiers for three configurations: VM_{MFCC} refers to using Mel-Frequency Cepstral Coefficients (MFCC) [39] calculated over the mixed voice and music signal, $VSep_{MFCC}$ refers to MFCC calculated over the automatically separated voice signal, and $VSep_{CPR}$ refers to using the cluster peak ratio (CPR), a feature that can be directly calculated on each extracted clusters of peaks. It is defined as:

$$CPR = \frac{\max(A^k)}{\text{mean}(A^k)} \tag{28}$$

where A^k are the extracted peak amplitudes for frame k. Voiced frames tend to have more pronounced peaks than unvoiced frames and therefore higher CPR values.

TABLE VII
VOICING DETECTION PERCENTAGE ACCURACY

	ZeroR	*NB*	*SVM*
VM_{MFCC}	55	69	69
$VSep_{MFCC}$	55	77	86
$VSep_{CPR}$	55	73	74

The experiments were performed using the Weka machine learning framework, where *NB* refers to a Naive Bayes classifier and *SVM* to a support vector machine trained using the sequential minimal optimization (SMO) [40]. The *ZeroR* classifier classifies everything as voiced and was used as a baseline. The goal was to evaluate the relative improvement in classification performance when using the separated voice signal rather than building an optimal voicing detector. No smoothing of the predictions was performed.

Table VII shows the classification accuracy (i.e the percentage of frames correctly classified using these 3 configurations). All the results were computed using 10-fold cross-validation over features extracted from the entire corpus. In 10-fold cross-validation the feature matrix is shuffled and partitioned into 10 "folds". The classification accuracy is calculated by using 9 of the folds for training and 1 fold for testing and the process is repeated 10 times so that all partitions become the testing set once. The classification results are averaged. 10-fold cross-validation is used to provide a more balanced estimate of classification accuracy that is not as sensitive to a particular choice of training and testing sets [40]. Using the automatically separated voice results in significant improvements in voicing detection accuracy. Additionally, the use of the simple and direct CPR feature still outperforms a more complex classifier trained on the mixed data.

IV. CONCLUSIONS AND FUTURE WORK

We described how spectral clustering using the *normalized cut* criterion for graph partitioning can be used for predominant melodic source separation. Our proposed method is based on a sinusoidal peak representation which enables close to real-time computation due to its sparse nature. Grouping cues based on amplitude, frequency and harmonicity are incorporated in a unified optimization framework. Harmonically-Wrapped Peak Similarity (HWPS), a novel harmonicity similarity measure was also proposed. Experimental results evaluating HWPS and the proposed method in the context of mixture separation, audio melody extraction and voicing detection were presented. The proposed algorithm is causal, efficient and doesn't require any prior knowledge or song-specific parameter tuning.

There are many possible directions for future work. Although not necessary for the operation of the algorithm, prior knowledge such as sound source models or score representations could easily be incorporated into the similarity calculation. For example the likelihood that two peaks belong to the same sound source model [15] could be used as an additional similarity cue. Additional cues such as common amplitude and frequency modulation as well as the use of timing information such as onsets are interesting possibilities for future work. Another interesting possibility is the addition

of common panning cues for stereo signals as proposed in [41], [42].

Limitations of the current method suggest other directions of future work. Even though some grouping of components corresponding to consonants is achieved by the amplitude and frequency similarity cues, the sinusoidal representation is not particularly suited for non-pitched sounds such as consonants sounds. Alternative analysis front-ends such as perceptually-informed filterbanks or sinusoids+transient representations could be a way to address this limitation.

In the current system, the cluster selection and resynthesis are performed for the entire song independently of whether a singing voice is present or not. The use of a singing voice detector to guide the resynthesis would surely result in better results. Implementing such a singing voice detector directly based on properties of the detected cluster is an interesting possibility. The resynthesis also suffers from artifacts that result from the limitations of the sinusoidal representation. An interesting alternative would be to retain the sinusoidal modeling front-end for grouping but use the entire STFT spectrum for the resynthesis of the extracted voice. As studied in [43], such a resynthesis stage is more flexible and reduces artifacts.

Finally, in the current implementation, clustering and cluster selection are performed independently for each "texture" window. In the future we plan to explore cluster continuity constraints (for example neighbouring clusters in time corresponding to the same source should have similar overall characteristics) as well as more sophisticated methods of cluster selection.

We believe our work shows the potential of spectral clustering methods for sound source formation and tracking. We hope it will stimulate more research in this area as it can have significant impact in many MIR applications such as singer identification, music transcription and lyrics alignment.

ACKNOWLEDGMENTS

The authors would like to thank Kirk McNally and Ewenn Camberlein for providing part of the testing corpus used for the experiments. In addition Luis Filipe Teixeira helped a lot with solving various coding problems related to the use of *Marsyas*. The authors would also like to thank Anssi Klapuri for kindly providing the multi pitch estimator code [37]. Finally, we would like to thank the anonymous reviewers for their thorough comments and suggestions for improvement.

REFERENCES

[1] F. Pachet and D. Cazaly, "A classification of musical genre," in *Proc. RIAO Content-Based Multimedia Information Access Conference*, 2000.

[2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 10, pp. 293–302, 2002.

[3] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high is the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[4] A. Ghias, J. Logan, D. Chamberlin, and B. Smith, "Query by Humming: Musical Information Retrieval in an Audio Database," *ACM Multimedia*, pp. 213–236, 1995.

[5] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo, "The MUSART testbed for query-by-humming evaluation," *Computer Music Journal*, vol. 28(2), pp. 34–48, 2004.

[6] Y. Kim, D. Williamson, and S.Pilli, "Towards quantifying the album effect in artist identification," in *Proc. of Int.Conf. on Music Information Retrieval (ISMIR)*, 2006.

[7] Y.Li and D. Wang, "Singing voice separation from monaural recordings," in *Proc. of Int.Conf. on Music Information Retrieval (ISMIR)*, 2006.

[8] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot, "One microphone singing voice separation using source-adapted models," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2005.

[9] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," in *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, London, UK, 2005.

[10] G. Poliner, D. Ellis, A. Ehmann, E. Gomez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, 2007.

[11] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[12] M. Lagrange and G. Tzanetakis, "Sound source tracking and formation using normalized cuts," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, USA, 2007.

[13] D. Rosenthal and H. Okuno, Eds., *Computational Auditory Scene Analysis*. Lawrence Erlbaum Associates, 1998.

[14] D. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*. Wiley, 2006.

[15] E. Vincent, "Musical source separation using time-frequency priors," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14(1), pp. 91–98, 2006.

[16] S. T. Roweis, "One microphone source separation," in *Proceedings of the Neural Information Processing Systems(NIPS)*, 2000, pp. 793–799.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22(8), pp. 888–905, 2000.

[18] D. Ellis and K. Lee, "Minimal-impact audio-based personal archives," in *Proc. ACM Workshop on Continuous Archival and Retrieval of Personal Experience (CARPE)*, New York, USA, 2004.

[19] R. Cai, L. Lu, and A. Hanjalic, "Unsupervised content discovery in composite audio," in *Proc. ACM Multimedia*, 2005.

[20] S. Dubnov and T. Appel, "Audio segmentation by singular value clustering," in *Proc. of Int.Conf. on Computer Music (ICMC)*, 2004.

[21] F. Bach and M. I. Jordan, "Blind one-microphone speech separation: A spectral learning approach," in *Proc. Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2004.

[22] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, vol. 7, pp. 1963–2001, 2006.

[23] S. Srinivasan and M. Kankanhalli, "Harmonicity and dynamics based audio separation," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'03)*, vol. 5, 2003, pp. 640 – 643.

[24] S. Srinivasan, "Auditory blobs," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'04)*, vol. 4, 2004, pp. 313 – 316.

[25] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on sinusoidal representation," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 34(4), pp. 744–754, 1986.

[26] M. Lagrange, S. Marchand, and J. Rault, "Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2007.

[27] M. S. Puckette and J. C. Brown, "Accuracy of frequency estimates using the phase vocoder," *IEEE Trans. on Audio and Speech Processing*, vol. 6 (2), 1998.

[28] S. Marchand and M. Lagrange, "On the equivalence of phase-based methods for the estimation of instantaneous frequency," in *Proc. European Conference on Signal Processing (EUSIPCO'2006)*, 2006.

[29] M. Lagrange and S. Marchand, "Estimating the instantaneous frequency of sinusoidal components using phase-based methods," *to appear in the Journal of the Audio Engineering Society*, 2007.

[30] M. Abe and J. O. Smith, "Design Criteria for Simple Sinusoidal Parameter Estimation based on Quadratic Interpolation of FFT Magnitude Peaks," in *117th Convention of the Audio Engineering Society*, San Francisco, October 2004, preprint 6256.

[31] R. Badeau, B. David, and G. Richard, "High resolution spectral analysis of mixtures of complex exponentials modulated by polynomials," *IEEE Trans. on Signal Processing*, vol. 54(4), pp. 1341–1350, 2006.

[32] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Proc. ICASSP*, vol. 2, 2000, pp. 765–768.

[33] J. Rosier and Y. Grenier, "Unsupervised classification techniques for multipitch estimation," in *116th Convention of the Audio Engineering Society*, 2004.
[34] L. Martins and A. Ferreira, "PCM to MIDI transposition," in *Proc. of Audio Engineering Society (AES)*, 2002.
[35] M. Lagrange and S. Marchand, "Assessing the quality of the extraction and tracking of sinusoidal components: Towards an evaluation methodology," in *Proceedings of the Digital Audio Effects (DAFx'06) Conference*, 2006, pp. 239–245.
[36] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. Springer, 2006.
[37] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *International Conference on Music Information Retrieval (ISMIR)*, Victoria, BC, Canada, 2006.
[38] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (version 4.5.06)," Retrieved December 13, 2006, from http://www.praat.org/.
[39] S. Davis and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357–366, Aug. 1980.
[40] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, 2005.
[41] A. Jourjine, S. Richard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures," in *Proc. ICASSP*, 2000.
[42] C. Avendano, "Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppression, and re-panning applications," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2003.
[43] M. Lagrange, L. G. Martins, and G. Tzanetakis, "Semi-Automatic Mono to Stereo Up-mixing using Sound Source Formation," in *122th Convention of the Audio Engineering Society*, Vienna, May 2007.

Mathieu Lagrange (M'07) was born in Caen, France, in 1978. He received the M.Sc. degree in computer science from the University of Rennes 1, Rennes, France, in 2000. He obtained a post-graduate diploma focusing on spectral sound synthesis at the University of Bordeaux 1, Talence, France. He carried out research on sound analysis and coding at France Telecom Laboratories in partnership with the LaBRI (Computer Science Laboratory) where he received the Ph.D. degree in 2004. He is currently a Research Assistant with the Computer Science Department, University of Victoria, Victoria, BC, Canada. His research focuses on structured modeling of audio signals applied to the indexing, browsing, and retrieval of multimedia.

Luis Gustavo Martins received his Diploma degree in Electrical and Computer Engineering in 1997, and completed a M.Sc. in Electrical and Computer Engineering in 2002, both from the University of Porto, Portugal. His M.Sc. thesis was on the topic of polyphonic music transcription and he is currently enrolled in a Ph.D. program in Electrical Engineering at the University of Porto, with a Ph.D grant from the Portuguese Foundation for Science and Technology (FCT). He is a researcher at the Multimedia and Telecommunications Unit of INESC Porto, where he carries out research on the areas of software development, signal processing, machine learning and audio content analysis. He has been actively involved in the European NoE projects VISNET I and VISNET II where he currently contributes with research on audio and music analysis. He is an active developer of the open source Marsyas audio processing software framework.

Jennifer Murdoch (S'06) received the B.CSc. degree from Dalhousie University, Canada, and is currently pursuing the M.Sc. degree in Computer Science at the University of Victoria, Canada. Her research interests are in the areas of signal processing, machine learning, and user interfaces for audio content analysis, as well as multimedia approaches to Computer Science education.

George Tzanetakis (S'98-M'02) received his B.Sc degree in computer science at the University of Crete, Greece and his M.A and Ph.D degrees in computer science from Princeton University. His Ph.D work involved the automatic content analysis of audio signals with specific emphasis of large music collections.

In 2003 he worked as a Post-Doctoral Fellow at Carnegie Mellon University on query-by-humming systems, polyphonic audio-score alignment and video retrieval. Since 2004 he is Assistant Professor of Computer Science (also cross-listed in Music and Electrical and Computer Engineering) at the University of Victoria, Canada. His research deals with all stages of audio content analysis such as analysis, feature extraction, segmentation, classification with specific focus on Music Information Retrieval (MIR). His work on musical genre classification is frequently cited and received an IEEE Signal Processing Society Young Author Award in 2004. He is the principal designer and developer of the open source Marsyas audio processing software framework.