Music Source Separation

Hao-Wei Tseng
Electrical and Engineering System
University of Michigan
Ann Arbor, Michigan
Email: blakesen@umich.edu

***Abstract*—In popular music, a cover version or cover song, or simply cover, is a new performance or recording of a previously recorded, by someone other than the original artist.**

However, it is impossible to retrieve a piece of single track for most of people. Therefore, my goal is to deliver a program that separates a record into several tracks, each corresponding to a meaningful source, which can be used for cover artists to facilitate their performance.

I. Introduction

Cover artists on Youtube have recently become increasingly popular. However, in order to make cover music, these artists have to acquire partial records. For example, a cover singer would sing with an off vocal version of the song; an accompaniment artist would play with a particular instrument removed from the original performance. Some off vocal tracks are released with the albums, which makes easy to acquire. However, in most cases, popular songs are not released with an off vocal version. Furthermore, tracks performed without certain instruments are hardly found in public market. These tracks are sometimes available in special cases. In result, most cover artists have to come up with their own solutions. One way to do so is to generate every track of a piece of music. This requires fundamental training in music, which is inaccessible to major public. As a result, I am going to provide a program which is able to separate the vocal and off-vocal tracks out.

II. Background & Difficulties

In my experiment, I focus on a solo singer with multiple instruments. Fig.1 Fig.2 That is, I assume my music pieces have no more than one vocal components with none or several off-vocal components in it. When looking onto the figures, the plot of time domain Fig.1looks like noise that has little use for my experiment. Therefore, I am going to analyze the signal of its frequency domainFig.2. It looks like the signal are mixed in the center. However, it's hard to tell vocal from off-vocal part, if I do the analysis on it directly. Worst of all, the frequncy of vocal and off vocal must have a portion of overlap. It makes it impossible to use a filter to separate the two part out. Fortunately, I can use one of the machine learning techniques for this problem Blind Source Separation(BSS) BSS is a useful and power technique for this kinds of problem. It is a technique to separate different sources from a set to mixtures without the prior knowledge of the source nor the way they are mixed. With this advantage, BSS is one of the most powerful algorithms to my problem. I include Independent Component Analysis(ICA) and Degenerate Unmixing Estimation Technique(DUET) for this project.

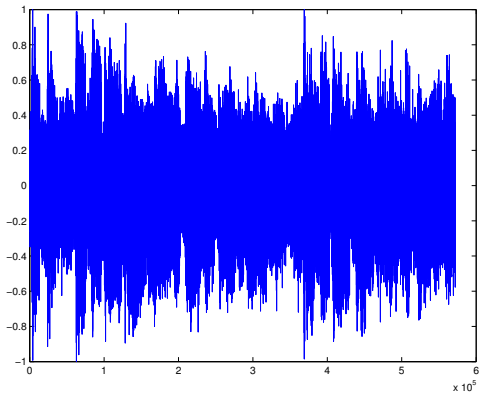

Fig. 1. Time domain of a music piece

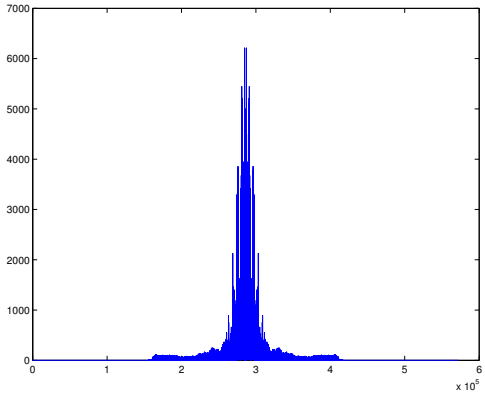

Fig. 2. Frequency domain of a music piece

A. ICA

ICA finds the independent components by maximizing the statistical independence of the estimated components. As a result, ICA is one of the most popular method in BSS, and is known for its application to separate mixtures of speech signals by taking the advantage of tracking the potential components blindly. I applied the FastICA toolbox provided by [1]. However, the number of output sources is limited by this approach (as formula below).

$$\hat{s} = Wx \simeq A^{-1}x$$

ICA needs more observations than independent components. But still, this algorithm is really good to separate the vocal and off-vocal. According to the formula, ICA can separate out estimated independent sources, which is no more than the numbers of observation that I provided, left voice and right voice. As a result, the output are like vocal and off-vocal part respectively. But still it contains some noise.

B. DUET

DUET separates degenerate mixtures is by partitioning the time–frequency representation of one of the mixtures. In other words, DUET assumes the sources are already separate in the time-frequency plane, the sources are disjoint. The demixing process is then simply a partitioning of the time–frequency plane. Although the assumption of disjointness may seem unreasonable for simultaneous speech, it is approximately true. By approximately, it means that the time-frequency points which contain significant contributions to the average energy of the mixture are very likely to be dominated by a contribution from only one source. Stated another way, two people rarely excite the same frequency at the same time. In this assumption, I can separate sources into several pieces.

A blind source separation problem is considered degenerated when the number of observations is less than that of the actual sources. In this sense, it is able to be used to separate more components out from the pieces. Traditional separation techniques such as ICA cannot solve such problems. However, DUET can blindly separate an arbitrary number of sources given just two anechoic (non-echonic) mixtures provided the time-frequency representations of the sources do not overlap too much [3]. With this advantages, DUET is able to separate more components out with better quality.

In some sources, by implement [2], it provides a good result like 4. In 4, the sources are two pieces of record of speech and result is perfectly estimated the speech components. It is able to assume that the speech components are well-anechoic. However, if it is not, that is, if the sources are mixture of instruments or with vocals, the output would be less usable and less acceptable 5 6. In 5 6, these two figures imply that DUET algorithm performs worse when sources are mixed of vocal and off-vocal tracks. When it is pure off-vocal part, as 6, there are two less mixing components in the plot (as noted by cursor), which are exactly two drums sources as checking manually. For 5 , the plot contains several different pulses, which shows the drawbacks of DUET.

C. CQT

Constant-Q Transform (CQT) has the similar idea as Fourier transform, but CQT is a logarithm scale of Fourier transform [4]. The following is the definition of CQT, where x[n] is the time domain signal, X[n] the frequency domain coefficient.

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]x[n]e^{\frac{-j2\pi Qn}{N[k]}}$$

W is a window function used to reduce aliasing effects near the maximum frequency. it is also used to isolate the signal to a short time period. The parameters are defined as the following.

$$N[k] = \frac{f_s}{\delta f_k} = Q\frac{f_s}{f_k}, \delta f_k = (2^{\frac{1}{b}})^k f)min$$

f_s is the sample rate and f_{min} is the minimum frequency. f_k is the center frequency of the kth coefficient. As the Discrete Fourier Transform(DFT) can be viewed as a series of filter banks, CQT can also be viewed as a series of exponentially spaced filters. In contrary to the linear resolution Fourier transform has, CQT has logarithm resolution in the frequency domain. Since the musical notes are spaced exponentially across each octave, CQT can linearly map the musical scales. This provided me an alternative way to map musical signals onto the time-frequency domain so that the instruments do not overlap too much.

I would like to implement an iterative CQT-based source separation algorithm to identify each instrument in an excerpt. Fig.3 shows the system diagram of the algorithm.

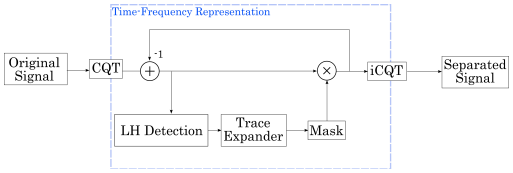

Fig. 3. System diagram of the proposed algorithm.

First, transform the input signal into the time-frequency domain by short-time CQT. This results in a spectrum of the original signal. The lowest harmonic within each timeslot is then traced on the spectrum. Once I locate the lowest harmonic, I expand and isolate the spectrum around those harmonic. This is called trace expansion. I then cluster the power spectrum of the trace. The lowest frequency cluster is extracted as the first instrument. After removing the signal of the first instrument from the observation, I repeat the whole procedure until no more instrument can be extracted. I use a bandpass filter for extraction.

III. Result

A. ICA

I chose several song excerpts of different genre as input, each lasting about 10 seconds. The two channels are passed as the observation to the FastICA toolbox. The output contains 2 separated signals. By listening, it is able to identify one source as the off-vocal version[5] of the original excerpt[6]. Most of the vocal parts are removed. The other source contains the vocal parts and some accompaniments. There is little distortion in both separated signals.

B. DUET

Fig.4 shows the time-frequency representation of a record provided by with 4 people speaking concurrently [7]. It obvious to identify that there are 4 disjoint peaks in the histogram, and, as expected, the corresponding reconstruction[8] of the 4 sources is clearly understandable.

Fig.5 shows the time-frequency representation of a pop music [6] excerpt. The histogram is more spread than that

of a speech signal, or I can say, they are less disjoint in this representation. The result is a poorer quality of reconstruction. I choose the four largest peaks as the center of the mask. The reconstructed signals, containing a lot of distortion noise, are hardly identifiable by human ears. Only the signal filtered from the main peaks contains recognizable voice.

Fig.6 shows the time-frequency representation of an electronic music[9] excerpt, where no voice presents. There are two peaks and hence two sources. The reconstructed signals identify the side drum and the [10] respectively. The sound of the rest of the instruments is still highly distorted in the 2 separated signals.

While DUET separates speeches from different people successfully, it performs poorly on separating vocal signal from the accompaniment and separating different instruments. DUET relies on the sources to be disjoint in the time-frequency domain, which is generally true for speech signals. However, this is not true for musical performances. In speech signals, only vowels contain concentrated power and consonants are merely white Gaussian noise, which has no significance in the frequency domain. Furthermore, vowels do not appear continually, resulting a highly disjoint time-frequency representation. On the other hand, musical instruments are often played continually, and moreover, the frequency components are much more complicated. Pitched musical instruments are often based on an approximate harmonic oscillator such as a string or a column of air, which oscillates at numerous frequencies simultaneously. The signal power are spread in each octave, giving a wide spread spectrum overlapping each other in the time-frequency representation. This also explained why the drums are separable by DUET. Since they are not pitched and percussion instruments are not played continually, they resemble speech signals in the time-frequency representation.

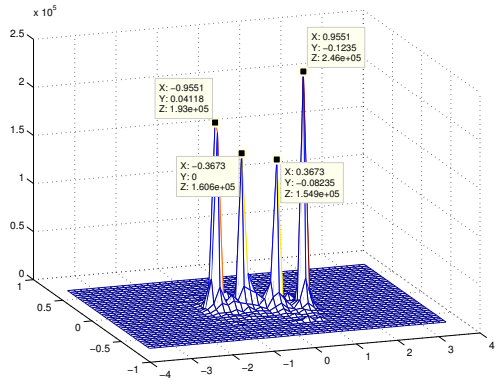

Fig. 4. Estimated independent components of speech by DUET

C. CQT

Fig.7 shows the time-frequency representation of a classical excerpt [11]. It can be recognized there are more than three major instruments and other harmonic wave. My method is to filter out each main instrument by tracing the energy, and then use k-means to cluster and select the result. For example, the

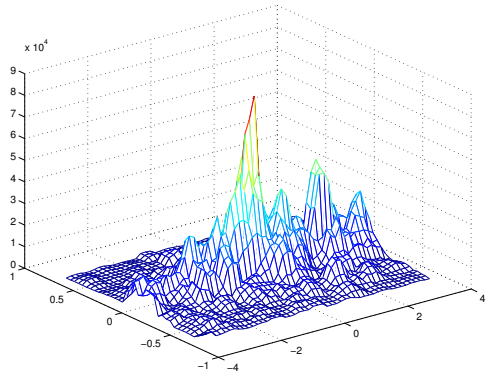

Fig. 5. Estimated independent components of pop music by DUET

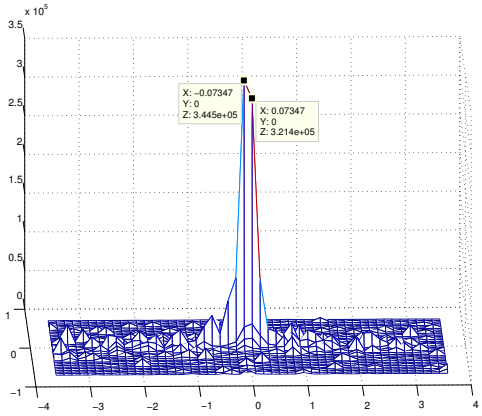

Fig. 6. Estimated independent components of electronic instruments by DUET

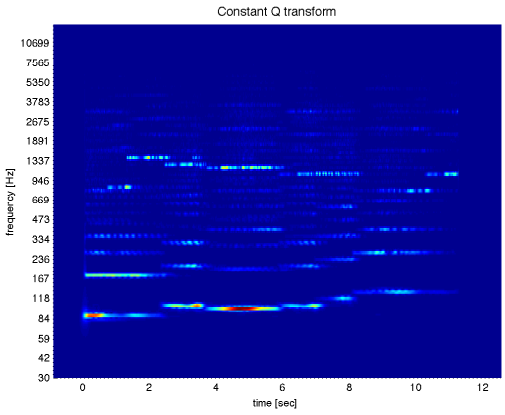

Fig. 7. Spectrum of the original classical music except by CQT

filter of the first estimation is like Fig. 8 and the corresponding result is Fig. 9.

Fig.10 shows the time-frequency representation of an estimated double bass [12]. I can tell this is the second trace corresponding to the original plot. There are some harmonic

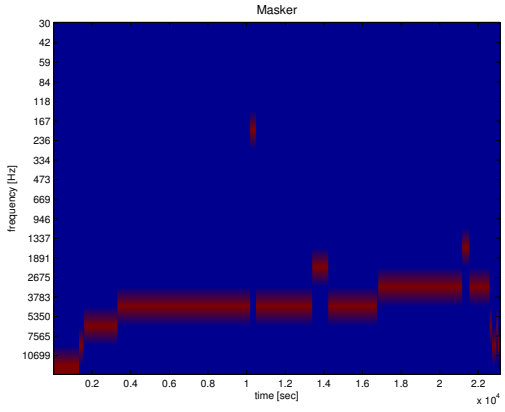

Fig. 8. First mask

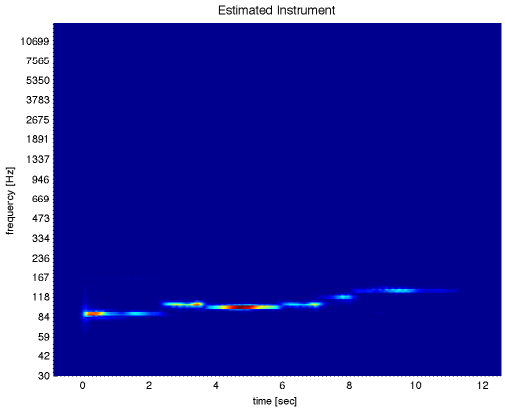

Fig. 9. First estimated instrument

waves, which is the genre of overtone ejected by double bass. By listening, it is clear double bass sound without distortion.

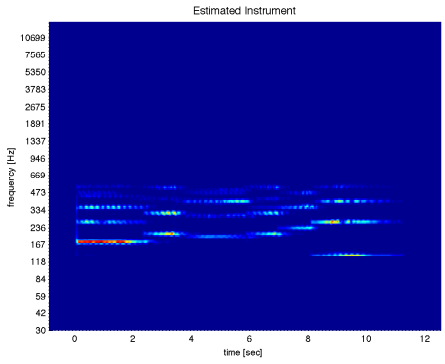

Fig. 10. Spectrum of a separated instrument "double bass"

Fig.11 shows the time-frequency representation of an estimated flute [13]. It can be recognized this is the top trace corresponding to original plot. There are some harmonic waves, which is overtone both from itself and other instruments. By listening, it is a flute sound with a little distortion, which might

TABLE I. SNR COMPARISON OF DUET AND PROPOSED ALGORITHM

	SNR of source 1	SNR of source 2
Proposed algorithm	5.46dB	4.65dB
DUET	1.9dB	-5.3dB

be caused by the distortion of overtone of other instrument.

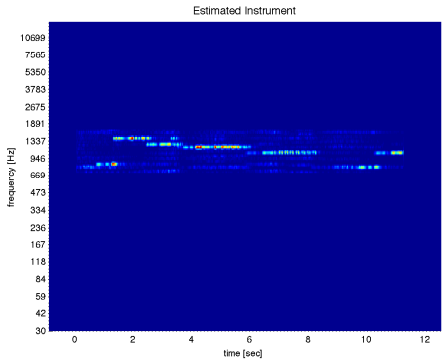

Fig. 11. Spectrum of a separated instrument "flute"

I also evaluated separation algorithm by comparing the signal-to-noise ratio(SNR) of the reconstructed sources of each separation techniques. Since the original tracks of each instrument in commercial releases are unavailable, generate a short piece of music [14] with 2 instruments for this experiment. Table I shows the SNR of the reconstructed signal. The noise defined to be the distortion of the reconstructed source. My algorithm is about 3dB better than DUET only, showing that CQT can better capture the features of musical instruments.

IV. CONCLUSION

ICA separates vocal and accompaniment successfully. However, ICA requires more observations than the number of sources. In my case, the observations are the two channels, left and right, of the track. This limits the output to two sources. If I wish to separate more sources, for example, different instruments in the accompaniment, I will need to exploit more features from the given source.

The DUET algorithm can separate an arbitrary number of sources given two anechoic observations [2]. However, it assumes that the sources are distinguishable in a time-frequency domain found by applying Fourier transform. This is true for speech signals where signal power is concentrated where vowels appears since consonants act as Gaussian white noise. However, for musical instruments, signal power is separated in each octave, which makes it hard to distinguish from one another.

CQT is another mapping from the time domain to the frequency domain. Unlike Fourier transform, CQT has a logarithm spacing in the frequency domain, giving it a linear representation of musical notes. This allows me to separate different musical instruments. I implemented an iterative method to isolate each sources from the time-frequency domain generated by CQT. My algorithm can separate different musical instruments from a given mixture, and has improved SNR of the estimated sources by 3dB compared to the original DUET.

To sum up, the tools from class that I used in this job are sampling to sample data from continuous time into discrete time; Fourier transforms, fast Fourier transform to transform my dataset into frequncy domain; filter designs with moving averages for extracting the estimated components. Additionally, I did this project with some machine learning techniques, such as ICA, DUET, k-means and CQT. Therefore, I acquire tons of knowledge in this project that gives me an opportunity to do some practical things.

V. Future Works

At the end of this project, I haven't succeeded to incorporate CQT with DUET as Fig 12. Instead, I implemented several different separation criteria in the time-frequency domain to exploit CQT to separate musical instruments. While I hand tune the masking parameters, machine learning techniques can be applied to learn the optimal clustering parameters in the frequency domain. Such techniques can also be incorporated with the DUET algorithm to automate peak detection.

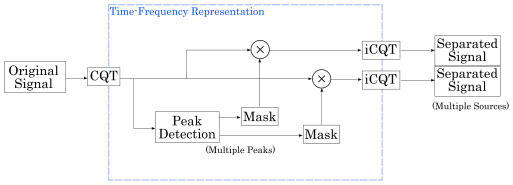

Fig. 12. System diagram of the proposed algorithm.

References

[1] http://research.ics.aalto.fi/ica/fastica/

[2] Scott Rickard, *The DUET Blind Source Separation Algorithm* , pages 217241. Springer Netherlands, 2007.

[3] Zafar Rafii and Bryan Pardo, "Degenerate Unmixing Estimation Technique using the Constant Q Transform," *36th International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic*, May 2227, 2011.

[4] Benjamin Blankertz, *The Constant Q Transform*, http://wwwmath.uni-muenster.de/logik/Personen/blankertz/constQ/constQ.html

[5] The music source is at musics/pop_offvocal.

[6] The music source is at musics/pop_origin which is found from youtube.

[7] The music source is at musics/speech which is found from http://eleceng.ucd.ie/~srickard/bss.html.

[8] The music source is at musics/speech_estimate.

[9] The music source is at musics/elec_origin which is found from youtube.

[10] The music source is at musics/elec_estimate.

[11] The music source is at musics/instrus_origin http://www.zafarrafii.com/research.html.

[12] The music source is at musics/instrus_bass.

[13] The music source is at musics/instrus_flute.

[14] The test_mixture is the mixture of test_input1 and test_input2; the corresponding output of DUET is test_duet_est1 and test_duet_est2, and the output of CQT is test_CQT_est1 and test_CQT_est2.