Musical Audio Source Separation Based on User-Selected F0 Track[*]

Jean-Louis Durrieu and Jean-Philippe Thiran

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Signal Processing Laboratory (LTS5)
Switzerland
firstname.lastname@epfl.ch[**]

Abstract. A system for user-guided audio source separation is presented in this article. Following previous works on time-frequency music representations, the proposed User Interface allows the user to select the desired audio source, by means of the assumed fundamental frequency (F0) track of that source. The system then automatically refines the selected F0 tracks, estimates and separates the corresponding source from the mixture. The interface was tested and the separation results compare positively to the results of a fully automatic system, showing that the F0 track selection improves the separation performance.

Keywords: User-guided Audio Source Separation, Graphical User Interface, Non-negative Matrix Factorization

1 INTRODUCTION

Most audio signals are mixtures of different sources, such as a speaker, an instrument, or noise. Applications such as speech enhancement or musical remixing require the identification and the extraction of one such source from the others.

While many existing musical source separation algorithms aim at blindly separating all the different instruments, the aim of the proposed system is to separate the source defined by the user. Let $\{x_t\}_{t=1\dots T}$ be a single-channel mixture signal of duration T. Let $\{v_t\}_t$ and $\{m_{r,t}\}_t$ respectively be the mono signals of the source of interest, usually a singing voice, and of the R remaining sources, *i.e.* the musical accompaniment. These signals are mixed such that:

$$x_t = v_t + \sum_{r=1}^{R} m_{r,t} \tag{1}$$

The task at hand is to estimate the signal of interest v_t, given user-provided information on the corresponding source. We propose a separation system that

[*] J.-L. Durrieu; e-mail: jean-louis.durrieu@epfl.ch

[**] This work was funded by the Swiss CTI agency, project n. 11359.1 PFES-ES, in collaboration with SpeedLingua SA, Geneva, Switzerland.

allows the user to choose the source in an intuitive way, thanks to a representation of the polyphonic pitch content of the audio excerpt. The system was tested by several users on a SiSEC 2011 [10] data set, and the contribution of the users is shown to improve the separation performance compared to the automatic system in [3].

This paper is organized as follows. The relevance of user-guided source separation is first discussed, followed by the presentation of the proposed Graphical User Interface (GUI). The underlying signal model, representation and the algorithm for source separation, mostly derived from previous works from the authors [3], are then briefly stated. The separation guided by the users is thereafter discussed and compared with the automatic separation system. Finally, we conclude with perspectives for the proposed system and concept.

2 User-Guided Source Separation

2.1 Related Works

Audio source separation methods essentially mimic auditory abilities: a human being can focus on the individual instruments of a mixture thanks to their locations, energies, pitch ranges or timbres. With multi-channel signals, such as stereo signals, one can infer spatial information [2], or train models to extract specific sources, even with single-channel signals [1].

The user can be required to provide some meta-information, such as the instrument name in a supervised framework [13], a musical score [5], the time intervals of activity for each instrument [7] or a sung target sound [11]. Musical scores or correct singing are however difficult to acquire, and are often not aligned with the mixture signal.

Expert users can be asked to choose the desired source through its position [14] or selecting components that are played by the desired instrument, thanks to intermediate separation results [15]. In [8], the automatically estimated melody line can be corrected by the user.

2.2 F0-guided Musical Source Separation

For musical audio excerpts, in particular for vocal sources, many studies have shown the relevance of the fundamental frequency (F0) contours. In [5], the authors use the music score to extract the notes, which helps estimating the actual F0 line of the instrument to remove. In [9], an estimated F0-contour is used to separate the corresponding instrument.

The goal of this work is to study to what extent user input can improve the separation of a specific source. Indeed, some ill-posed issues in the automatic separation problem, using F0 contours, can arise. First, with many interfering sources, it is difficult to automatically decide whether a specific source is present or not. Furthermore, octave and other harmonic-related confusions in the F0 representation can lead to erroneous separations. These errors may easily be corrected by a trained user who uses the context to solve these ambiguities.

3 Graphical User Interface

3.1 Ergonomy issues

Allowing the user to dynamically choose the desired source requires a representation that clearly displays the possible choices. The waveform would not allow to locate, in time and in frequency, sources that are overlapping in time. Time-frequency representations (TFR), such as the short-term Fourier transform (STFT), are therefore required to visually identify such sources. With a time x-axis and a frequency y-axis, the sinusoids (horizontal lines) or the noises (vertical patterns) corresponding to the desired source easily stand out. Such an approach would however require a significant amount of work, and would not scale well.

Harmonic sources exhibit a characteristic graphical pattern, in the STFT, for each F0: the system in [3] identifies these patterns and provides the energy of the different F0s for each signal frame. From such a representation, the user can select the desired source thanks to its melody line, with little effort.

Furthermore, representing the pitch on the Western musical scale is a visualization that many users can understand. For instance, in [6], Klapuri proposes such a "piano-roll" visualization.

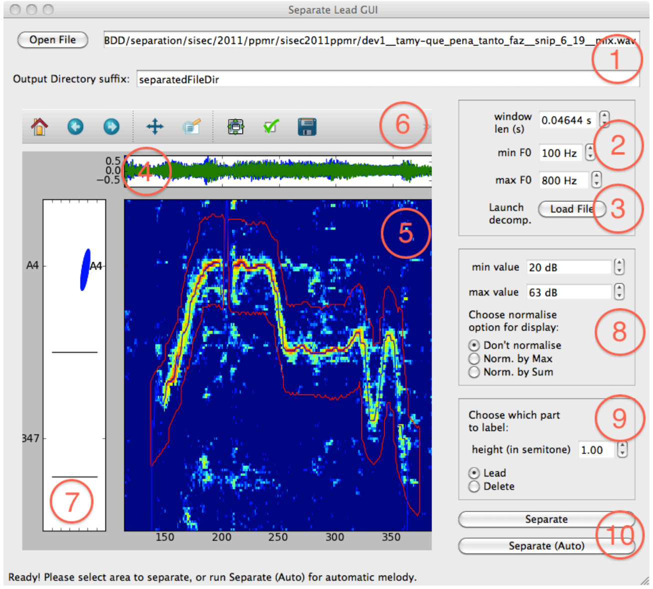
In this article, the mid-level representation introduced in [3] was chosen, because it is easy to configure so as to look like a piano-roll. The method however relies on a fixed dictionary of harmonic spectral shapes, and the proposed system is therefore better suited for the separation of corresponding sources, such as wind instruments, voice or bowed string instruments.

3.2 Practical solutions

Using Python/NumPy, with the Matplotlib and PyQt4 modules [12], it was possible to design a GUI taking advantage of the representation in [3].

A screen capture of the proposed GUI application is shown on Fig. 1, with the following elements: (1) specify the audio file and the output folder, (2) parameter controls, for the analysis window length, the minimum and maximum candidate F0, (3) a button to "load the file" (computing the decomposition of Sect. 4.1), (4) the waveform of the audio file, (5) the energies for each frame and for each F0 candidate, on which the user can select the melody F0 track (time on x-axis and F0 on y-axis), (6) a toolbar, for zooming and exploring, (7) a representation (musical staff) to indicate the corresponding F0s or notes, (8) normalization choices for the image, (9) buttons toggling between selection ("Lead") and deselection ("Delete"), plus a field to choose the vertical extent of the selection (in semitone), (10) "Separate" and "Separate (Auto)" buttons to launch the separation with or without the user selected track, respectively.

The user can select on (5) a region and thus identify it as a desired F0 range. Once she is finished with her choice, she can start the separation with one of the "Separate" buttons. The underlying mechanisms are further explained in the following section.



Open File: BDD/separation/sisec/2011/ppmr/sisec2011ppmr/dev1__tamy-que_pena_tanto_faz__snip_6_19__mix.wav

Output Directory suffix: separatedFileDir

window len (s): 0.04644 s

min F0: 100 Hz

max F0: 800 Hz

Launch decomp.: Load File

min value: 20 dB

max value: 63 dB

Choose normalise option for display:
- ● Don't normalise
- ○ Norm. by Max
- ○ Norm. by Sum

Choose which part to label:

height (in semitone): 1.00

- ● Lead
- ○ Delete

Ready! Please select area to separate, or run Separate (Auto) for automatic melody.

Fig. 1. GUI for selecting the desired F0 track.

4 F0 Representation and Separation Algorithm

The audio signal model presented in [3] is first briefly described. The computation of the F0 representation is then discussed, and at last the user-assisted separation algorithm of the selected source is presented.

4.1 Audio Signal Model

The audio mixture is modelled through its $F \times N$ short-term power spectrum (STPS) matrix $\mathbf{S}$, defined as the power of its STFT $\mathbf{X}$, with F the number of Fourier frequencies and N the number of frames. For simplicity, the model is presented for the single-channel case, but the stereo model of [3] was used for the experiments of this article.

$\mathbf{S}$ is assumed to be the sum of the STPS of the signal of interest $\mathbf{S}^V$ with the residual STPS $\mathbf{S}^M$:

$$\mathbf{S} = \mathbf{S}^V + \mathbf{S}^M \tag{2}$$

$\mathbf{S}^V$ is the element-wise product of a "source" part (F_0) by a "filter" part ($\varPhi$):

$$\mathbf{S}^V = \mathbf{S}^{\varPhi} \bullet \mathbf{S}^{F_0} \tag{3}$$

All the contributions $\mathbf{S}^{\varPhi}$, $\mathbf{S}^{F_0}$ and $\mathbf{S}^M$ are further modelled as non-negative matrix products of a spectral shape matrix ($\mathbf{W}^{\varPhi}$, $\mathbf{W}^{F_0}$ and $\mathbf{W}^M$, with K, U

and R elementary shapes, respectively) by the corresponding amplitude matrix ($\mathbf{H}^{\Phi}$, $\mathbf{H}^{F_0}$ and $\mathbf{H}^{M}$). Finally:

$$\mathbf{S} = \mathbf{W}^{\Phi}\mathbf{H}^{\Phi} \bullet \mathbf{W}^{F_0}\mathbf{H}^{F_0} + \mathbf{W}^{M}\mathbf{H}^{M} \tag{4}$$

In (4), all the parameters of the right hand-side are estimated on the signal, except the matrix $\mathbf{W}^{F_0}$ which is a dictionary of harmonic spectral "comb", parameterized by its F0 frequency. As discussed in [3], a careful choice of the F0s used in that dictionary leads to the desired representation in $\mathbf{H}^{F_0}$: in our case, we chose $\log_2$-spaced F0 values, *i.e.* a scale proportional to the Western musical scale. The number of F0s per semitone is fixed to 16, and the user can choose the extents of the scale, to fit the expected tessitura.

The other parameters are estimated thanks to the Non-negative Matrix Factorization (NMF) algorithm developed in [3]. The resulting matrix $\mathbf{H}^{F_0}$ finally provides the user with an image in which high values correspond to high energies associated with F0 frequencies, as shown on Fig. 1.

4.2 F0 line selection and usage

The user can then, through the GUI of Fig. 1, select the zones containing the F0 values that correspond to the desired melody. A binary mask matrix $\mathcal{H}$, of the same size as $\mathbf{H}^{F_0}$, initialized to 0 everywhere, is updated each time the user draws a curve with the mouse (while holding the left button) over the $\mathbf{H}^{F_0}$ image. All the coefficients along that curve, as well as the coefficients located within a user-defined vertical extent (half a semitone by default) are set to 1. The program superimposes the contour of the selection on the $\mathbf{H}^{F_0}$ image.

Once all the desired tracks have been selected, the user can trigger the separation, given her mask $\mathcal{H}$. Let $\widetilde{\mathbf{H}}^{F_0} = \mathcal{H} \bullet \mathbf{H}^{F_0}$. Assuming the desired source generates smooth melody lines, the melody path is then tracked in $\widetilde{\mathbf{H}}^{F_0}$ with a Viterbi algorithm [4]: the user-defined regions are therefore used to restrict the melody tracking. The user can also refine the chosen regions with a narrower vertical extent, effectively allowing non-smooth melodies if needed.

Finally, the smoothed-out melody line is used to create a refined version of $\widetilde{\mathbf{H}}^{F_0}$, zeroing coefficients lying too far from the melody. The parameters are then re-estimated, using $\widetilde{\mathbf{H}}^{F_0}$ as initial $\mathbf{H}^{F_0}$ matrix. These updated parameters $\{\mathbf{H}^{F_0}, \mathbf{W}^{\Phi}, \mathbf{H}^{\Phi}, \mathbf{W}^{M}, \mathbf{H}^{M}\}$ are used to compute the separated sources. This second estimation round focuses on voiced patterns, and a third round is done to include more unvoiced elements [3].

4.3 Separating the Selected Source

Wiener filters are used to separate the sources, obtaining the estimates of the STFT $\mathbf{V}$ and $\mathbf{M}$, using [3]:

$$\widehat{\mathbf{V}} = \frac{\mathbf{W}^{\Phi}\mathbf{H}^{\Phi} \bullet \mathbf{W}^{F_0}\mathbf{H}^{F_0}}{\mathbf{W}^{\Phi}\mathbf{H}^{\Phi} \bullet \mathbf{W}^{F_0}\mathbf{H}^{F_0} + \mathbf{W}^{M}\mathbf{H}^{M}} \bullet \mathbf{X} \text{ and } \widehat{\mathbf{M}} = \mathbf{X} - \widehat{\mathbf{V}} \tag{5}$$

The time-domain signals are then retrieved using an inverse STFT (overlap-add procedure).

5 EXPERIMENTS

5.1 Database and Protocoles

In order to evaluate the usage and the performance of the proposed user-guided source separation system, the development set (5 excerpts) for the SiSEC 2011 "Professionally Produced Music Recordings" task [10] is used.

Three users were asked to try the software. They were all used to handling computer softwares and had some background knowledge in music. The representation and separation principle were explained to each user beforehand. They provided their feedback about the software usage, Sect. 5.2, and the separation scores are discussed in Sect. 5.3. All the systems and users discussed in this section used the same default following parameters: $K = 4$, $U = 577$ (for 16 F0s per semitone, from 100 to 800Hz), $R = 40$, $F = 1025$ (for Fourier tranforms of size 2048, *i.e.* 46.44ms@44.1kHz) and with 25 iterations of the NMF algorithm.

5.2 Usage Feedbacks

The users first tested an early version of the GUI, and their observations were mostly linked with ergonomy issues or missing features (audio feedback, better display). Following their recommendations, we refined the GUI such that the focus was turned to the usability of the F0 representation.

For "easy" songs, with a clearly voiced, sustained vocal track, the F0 representation makes it easy to choose the desired source. However, for near-spoken or weak sources, identifying the vocal tracks was felt as a difficult task: for instance, one user declared not to be able to proceed with two songs for this reason (marked as '-' in Tab. 1, user #3). In addition, it is interesting to note that other types of sources are also harder to locate (both in time and frequency) than vocals, such as guitar or piano tracks.

5.3 Separation Performance

Table 1. Source separation results, see text for details.

	Mix	Auto		#1		#2		#3		[7]	SiSEC [10]				
Song	-	V	U	V	U	V	U	V	U		S3	S4	S5	S6	S7
dev1_bearlin	-5.3	4.7	4.9	6.1	**6.2**	5.4	5.8	4.7	5.1	-	-	-	3.3	3.2	-
dev1_tamy	0.2	8.6	8.9	10.3	10.1	**10.7**	10.6	8.9	9.2	-	-	-	7.1	8.7	10.3
dev2_another	-3.0	5.1	5.7	5.7	6.2	5.8	**6.6**	-	-	-0.7	-2.9	-2.8	3.4	2.2	-
dev2_fort	-7.2	2.3	2.4	3.2	3.7	3.4	**3.8**	-	-	3.2	-	-5.9	2.5	2.5	-
dev2_ultimate	-7.5	3.2	3.4	4.0	**4.4**	3.8	4.0	3.2	3.5	2.6	-0.9	-10.2	-0.6	1.4	-

The Signal-to-Distortion-Ratios (SDRs) for the estimated vocal source for each user (#1, #2 and #3) are reported in Tab. 1. The results obtained when using the mixture x as the vocals estimation (Mix), when using the fully automatic system [3] (Auto), those of the algorithm in [7] (from the SiSEC website [10]) and the SiSEC 2011 results for 5 algorithms (S3 to S7) [10] are also given.

The SDRs for the "Auto" and user-guided systems are better than those of the other systems, even the other user-guided system [7]. Furthermore, these examples show that the system is able to take advantage of the user-provided information. Some songs might be more challenging, such as the rap song (dev2_fort), probably because the desired vocal signal is closer to speech than to singing voice. The inadequation of the chosen $\mathbf{H}^{F_0}$ representation for this type of sources was already discussed [3], and the present study shows that even trained users could hardly use it for these signals.

6 CONCLUSION

A novel user-guided audio source separation system is proposed, allowing the user to easily select a harmonic audio source she desires to separate from a musical audio mixture. The energy of different hypothesized F0 candidates is displayed. Once the user has selected the relevant F0 melody track, the system automatically finds the F0 path maximizing the energy within the regions of interest, estimates the corresponding source and separates it using Wiener filtering and NMF-derived techniques.

The proposed system delegates the source identification to the user, such that there is less ambiguity with the definition of the target source, for the system. The evaluation of the system therefore becomes more relevant. The chosen representation also allows the choice of the source to be straightforward, especially for songs, where the lead singer usually dominates the mixture, providing a fairly readable representation.

The system and GUI could be further improved by adding, for instance, partial separation excerpts allowing the user to listen to what specific chunks of the representation correspond to, before performing the final separation. The user may also want to identify sources from the musical background that are not to be included in the desired source. Such a feature would require to search how to integrate such a prior into the separation stage.

The technique could be used for other applications, such as speech enhancement. The extension to one-speaker signals is straightforward, but many-speakers signals lead to representations that are harder to interprete. Finally, the system could be used as annotation tool: it could assist semi-automatic transcription music signals into musical scores, where an automatic system would infer note boundaries, rhythms, key and time signature from the user inputs.

7 ACKNOWLEDGMENT

This work was funded by the Swiss CTI agency, project n. 11359.1 PFES-ES, in collaboration with SpeedLingua SA, Lausanne, Switzerland. The authors would also like to thank S. Arberet, F. de Morsier and L. Navarro for their help in testing the GUI.

References

1. Benaroya, L., Bimbot, F., Gribonval, R.: Audio source separation with a single sensor. IEEE Transactions on Audio, Speech and Language Processing 14(1), 191–199 (January 2006)
2. Cardoso, J.F., Souloumiac, A.: Blind beamforming for non Gaussian signals. IEE Proceedings-F 140(6), 362–370 (December 1993)
3. Durrieu, J.L., Richard, G., David, B.: A musically motivated representation for pitch estimation and musical source separation. IEEE Journal of Selected Topics on Signal Processing 5(6), 1180 – 1191 (October 2011)
4. Durrieu, J.L., Richard, G., David, B., Févotte, C.: Source/filter model for unsupervised main melody extraction from polyphonic audio signals. IEEE Transactions on Audio, Speech, and Language Processing 18(3), 564 –575 (March 2010)
5. Han, Y.S., Raphael, C.: Desoloing monaural audio using mixture models. In: Proceedings of the International Conference on Music Information Retrieval. Vienna, Austria (September 23 - 27 2007)
6. Klapuri, A.: A method for visualizing the pitch content of polyphonic music signals. In: Proceedings of the 10th International Society for Music Information Retrieval Conference. pp. 615–620. Kobe, Japan (October 26-30 2009)
7. Ozerov, A., Fevotte, C., Blouet, R., Durrieu, J.L.: Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation. In: proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 257 –260. Prague, Czech Republic (22-27 May 2011)
8. Pant, S., Rao, V., Rao, P.: A melody detection user interface for polyphonic music. In: Proc. of the National Conference on Communications. Madras, Chennai, India (January 29-31 2010)
9. Ryynänen, M., Virtanen, T., Paulus, J., Klapuri, A.: Accompaniment separation and karaoke application based on automatic melody transcription. IEEE International Conference on Multimedia and Expo pp. 1417–1420 (2008)
10. SiSEC: Professionally produced music recordings. Internet page: `http://sisec.wiki.irisa.fr/tiki-index.php?page=Professionally+produced+music+recordings` (2011)
11. Smaragdis, P., Mysore., G.: Separation by "humming": User-guided sound extraction from monophonic mixtures. In: Proceedings of IEEE Workshop on Applications Signal Processing to Audio and Acoustics. pp. 69 – 72 (October 18-21 2009)
12. Tosi, S.: Matplotlib for Python developers. Packt Publishers (2009)
13. Vincent, E.: Musical source separation using time-frequency source priors. IEEE Transactions on Audio, Speech and Language Processing 14(1), 91–98 (January 2006)
14. Vinyes, M., Bonada, J., Loscos, A.: Demixing commercial music productions via human-assisted time-frequency masking. In: Convention Paper, the 120th AES Convention. Paris, France (May 20-23 2006)
15. Wang, B., Plumbley, M.D.: Musical audio stream separation by non-negative matrix factorization. Proc. of the DMRN Summer Conference (July 23-24 2005)