SEPARATION BY "HUMMING": USER-GUIDED SOUND EXTRACTION FROM MONOPHONIC MIXTURES

Paris Smaragdis

Advanced Technology Labs
Adobe Systems Inc.

Gautham J. Mysore

Center for Computer Research in Music and Acoustics
Stanford University

ABSTRACT

In this paper we present a novel approach for isolating and removing sounds from dense monophonic mixtures. The approach is user-based, and requires the presentation of a guide sound that mimics the desired target the user wishes to extract. The guide sound can be simply produced from a user by vocalizing or otherwise replicating the target sound marked for separation. Using that guide as a prior in a statistical sound mixtures model, we propose a methodology that allows us to efficiently extract complex structured sounds from dense mixtures.

1. INTRODUCTION

A long running problem in systems that perform source separation, is that of specifying the sound one wants to separate. Varying source separation approaches treat this issue in a form that best fits their framework. Research on fully determined multichannel mixtures [1] is predominantly designed to extract all sources and sidesteps the issue, whereas undetermined multichannel approaches and beamforming tend to focus on the loudest sources, or sources emanating from a given direction [2, 3]. Single channel approaches [4, 5, 6] are usually designed to conform to a predefined sound model (e.g. a known speaker, or a harmonic series), and denoising approaches such as Wiener filtering rely on a known noise profile. In many situations, the above approaches do not pose significant problems when it comes to picking the right source. However in complex single-channel situations, such as the cases where one observes multiple similar sources, or cases where the sound of interest is not easy to model, one can often encounter difficulties pointing a separation routine towards the desired sound.

In this paper we present a novel approach in pointing an algorithm towards a target sound. We incorporate information provided by the user, who is expected to provide a sound that roughly mimics the desired source. For example, in order to separate a singer from a music recording, the user can simply sing or hum that part, or in the case of other sounds vocalize (or otherwise roughly replicate) the part of interest. In order to do this we present a statistical model for sound mixtures, and an appropriate priors framework which allows us to incorporate the user's guidance in the separation process. The resulting system is very efficient, provides the ability to extract hard to pinpoint sources, and as we will demonstrate produces very satisfactory results from difficult mixtures.

2. MODELING SOUND MIXTURES

In this section we describe the statistical model we use for our approach. We first define its basic formulation, and then present an extension which makes use of a priors structure which is necessary for the needs of this paper.

2.1. The PLCA Model

The model we choose to use is that of the Probabilistic Latent Component Analysis (PLCA) [6]. This is an additive model which is applied in the magnitude time/frequency domain and is a member of a wide family of similar models which have recently been very successful in decomposing mixtures of sounds [5, 6, 7, 8]. These models decompose magnitude or power spectrograms into sums of outer products of spectral and temporal components, which can be thought of as spectral bases and their corresponding weights. More specifically, in the case of PLCA, a magnitude spectrogram $\mathbf{F}$, is interpreted as a histogram that measures energy at every time/frequency cell and thus decomposed by a weighted series of products of marginal distributions along the frequency and time axes, i.e.:

$$\mathbf{F} \approx \gamma \sum_{z=1}^{M} P(z)P(f|z)P(t|z) \qquad (1)$$

The parameters $P(f|z)$ and $P(t|z)$ are distribution pairs along the frequency (f) and time (t) axes, conditional on a latent variable index z. The distribution $P(z)$ defines how these pairs are weighted to approximate the input, and the constant γ is a scaling factor. Effectively in $P(f|z)$ we learn frequency distributions that are used to construct the input mixture, and $P(t|z)$ represents how they appear in time. The constant M defines how many of these pairs we use

to approximate the input. In the case where $M = 1$, $P(f|z)$ ends up being the magnitude spectrum of the input, and $P(t|z)$ its magnitude envelope across time. The parameters in this model can be estimated using the Expectation-Maximization algorithm, which results in the following update equations:

$$\begin{aligned}
\text{E-step: } P(z|f,t) &= \frac{P(z)P(f|z)P(t|z)}{\sum_{z'} P(z')P(f|z')P(t|z')} \\
\text{M-step: } P(f|z) &= \frac{\sum_t \mathbf{F}_{f,t} P(z|f,t)}{\sum_{f'} \sum_t \mathbf{F}_{f',t} P(z|f',t)} \\
P(t|z) &= \frac{\sum_f \mathbf{F}_{f,t} P(z|f,t)}{\sum_f \sum_{t'} \mathbf{F}_{f,t'} P(z|f,t')} \\
P(z) &= \frac{\sum_f \sum_t \mathbf{F}_{f,t} P(z|f,t)}{\sum_{z'} \sum_f \sum_t \mathbf{F}_{f,t} P(z'|f,t)}.
\end{aligned}$$

2.2. Dirichlet Priors

Now let us introduce a mechanism to impose priors for the estimated parameters of this model. The distributions $P(f|z)$ and $P(t|z)$ in the PLCA model are multinomial distributions. As the Dirichlet distribution is a conjugate prior distribution to the multinomial distribution, it can be used to enforce our biases on the structure of the model distributions. The Dirichlet distribution is defined by a set of positive and real *hyperparameters* $\alpha(i)$. For our purposes, and with no loss of generality, we will assume that $\sum \alpha(i) = 1$ and we use an additional weight parameter to scale them arbitrarily. Doing so, we can then interpret the Dirichlet hyperparameters as a multinomial distribution that can serve as an exemplar for our estimates. The priors for all the frequency distributions Λ_f, and temporal distributions Λ_t, are:

$$\begin{aligned}
P(\Lambda_f) &\propto \prod_z \prod_f P(f|z)^{\kappa_z \alpha(f|z)} \\
P(\Lambda_t) &\propto \prod_z \prod_t P(t|z)^{\mu_z \alpha(t|z)}
\end{aligned}$$

where $\alpha(f|z)$ and $\alpha(t|z)$ are the "exemplar" hyperparameters. The weight scalars κ_z and μ_z can be interpreted as parameters expressing how much we wish to impose the priors. Using the above, the estimation equations for $P(f|z)$ and $P(t|z)$ now change to:

$$\begin{aligned}
P(f|z) &= \frac{\sum_t \mathbf{F}_{f,t} P(z|f,t) + \kappa_z \alpha(f|z)}{\sum_{f'} \sum_t \mathbf{F}_{f',t} P(z|f',t) + \kappa_z \alpha(f'|z)} \\
P(t|z) &= \frac{\sum_f \mathbf{F}_{f,t} P(z|f,t) + \mu_z \alpha(t|z)}{\sum_f \sum_{t'} \mathbf{F}_{f,t'} P(z|f,t') + \mu_z \alpha(t'|z)}.
\end{aligned}$$

3. USING SOUNDS TO SELECT SOUNDS

As shown in [6], the PLCA model can be used to separate sounds if it is provided with pre-trained components that can describe at least some of the sounds in a mixture. In the situation we examine here, we do not use pre-trained models and are provided only with a full mixture. In this section we show how we can use a user's guiding audio input to help us model the mixture, and then how we can isolate a desired target sound and the background mixture.

3.1. Modeling mixtures using an example

Our goal is to use a sound provided by a user, that can help define the target sound in the separation process. We will do that using the priors models introduced in the previous section. The overall separation process is described in the following steps:

- Record a sound $s_u(t)$ that mimicks the target sound in a mixture $s_m(t)$. There should be some audible similarities in both frequency and temporal behavior.
- Obtain the magnitude spectrograms $\mathbf{F}^u$ and $\mathbf{F}^m$ of $s_u(t)$ and $s_m(t)$ respectively.
- Estimate a M-component PLCA model with parameters $P_u(f|z)$ and $P_u(t|z)$ from $\mathbf{F}^u$.
- Estimate a $\{M+N\}$-component PLCA model with parameters $P_m(f|z)$ and $P_m(t|z)$ from $\mathbf{F}^m$.
  - Use $P_u(f|z)$ and $P_u(t|z)$ as hyperparameters $\alpha(f|z)$ and $\alpha(t|z)$ for the first M components of $P_m(f|z)$ and $P_m(t|z)$. Gradually decrease κ_z and μ_z from 1 to 0 throughout iterations.
  - Simultaneously learn the remaining N components of the model without using any priors.

What this process will result in, is M components that will describe the target sound, as defined by the user, and N components that will describe the rest of the mixture. If $s_u(t)$ is a fair approximation of the target sound, then the outlined procedure will slowly transform the components $P_u(f|z)$ and $P_u(t|z)$ to an appropriate $P_m(f|z)$ and $P_m(t|z)$ that model the target sound. The rest of the components of the mixture model, will not be biased to look like the target sound and will start explaining other parts of the input.

3.2. Separation process

Once we go through the process outlined in the previous section we will be left with a model which is segmented in two groups. Components in group $\mathcal{Z}_1 = \{1, ..., M\}$ of $P_m(f|z)$ and $P_m(t|z)$ will model the time/frequency energy of the target sound, and components in group $\mathcal{Z}_2 = \{M+1, ..., N\}$ will model the rest of the mixture. Knowing that, we can approximate the magnitude spectrogram contribution of the target by $\sum_{\mathcal{Z}_1} P_m(z) P_m(f|z) P_m(t|z)$, and the remainder by $\sum_{\mathcal{Z}_2} P_m(z) P_m(f|z) P_m(t|z)$. Since these two submodels are not guaranteed to explain all of the

original energy in the input magnitude spectrogram, we will instead use their posterior distributions which will distribute all of the original input's energy to both. In this case, we will compute the contribution of each component set using:

$$P(z \in \mathcal{Z}_1|f,t) = \frac{\sum_{z' \in \mathcal{Z}_1} P(z)P(f|z)P(t|z)}{\sum_{z' \in \{\mathcal{Z}_1,\mathcal{Z}_2\}} P(z')P(f|z')P(t|z')}$$

$$P(z \in \mathcal{Z}_2|f,t) = \frac{\sum_{z' \in \mathcal{Z}_2} P(z)P(f|z)P(t|z)}{\sum_{z' \in \{\mathcal{Z}_1,\mathcal{Z}_2\}} P(z')P(f|z')P(t|z')},$$

where $P(z \in \mathcal{Z}_1|f,t)$ and $P(z \in \mathcal{Z}_2|f,t)$ are essentially two soft masks for the target and the remainder of the mixture respectively. We can now modulate these two masks with the complex spectrogram of the original mixture, and then invert the results in order to obtain the two time-series for the target sound and the rest of the mixture. We can also add an additional binary masking step which in practice improves audible separation performance. We can compute a hard assignment for each time/frequency cell to each of the two resulting sounds. We do so by comparing the posterior likelihoods of the two groups and assigning each spectrogram element to the highest-likelihood group. We can additionally impose some frequency and temporal masking properties by convolving the posteriors with a Gaussian distribution. This results in increased suppression of the background sounds, although it can produce separation artifacts such a musical noise.

4. RESULTS

We now present results from a suite of experiments. We first validate that this approach can perform separation using synthetic mixtures simulations, and then we show a representative result from a real-world mixture.

4.1. Synthetic examples

In order to obtain some quantitative analysis of the performance of this algorithm, we run a series of synthetic monophonic mixtures of speech and music. We examined three distinct cases. The first case was the *oracle* test, in which the user provided speech utterance was the same as the target. The second case, used two different speakers of the same gender, and the third case used two different speakers of different genders. We additionally replicated the second case experiment by also using a binary mask as described in the previous section. In all these experiments we measured the Signal to Distortion Ratio (SDR), the Signal to Interference Ratio (SIR), and the Signal to Artifacts Ratio (SAR) as defined in [9]. The resulting values are shown in figure 1. As expected the oracle case if the best performer, producing audibly almost perfect results. When the user was the same gender as the one producing the target speech, the

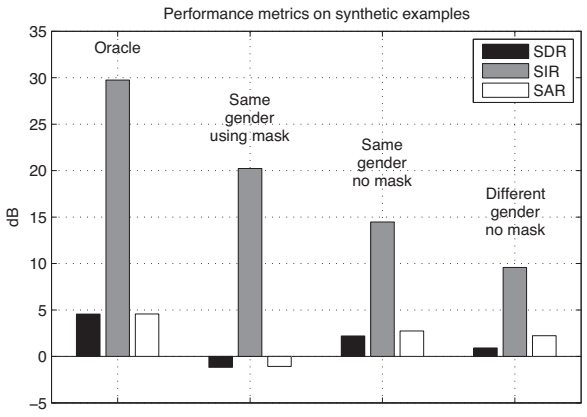

Figure 1: Performance metrics for synthetic example data of speech on music. We show averaged results from 50 runs of four distinct experiments. The leftmost set of bars shows the oracle performance, in which the guide sound was the same as the target. The second plot shows the case where the guide sound was the same gender as the target sound, and binary masking was used. The third example shows the same case but without the binary masking, and the rightmost bars show the case where the guide and target sounds were substantially different.

results were somewhat worse since there was less overlap in tonal character and in timing. We see this exaggerated even more so when the user/target genders were different. The use of a binary mask results in an interesting behavior. We see a significant increase in the SIR performance, but much worse scores in SDR and SIR. The increase in SIR is due to the fact that the interfering sound is suppressed much more abruptly since we completely zero-out some of its parts, on the other hand that process introduces distortions and musical noise which are to blame for the worse SDR and SAR values.

4.2. Results on real recordings

We also performed a variety of tests on music tracks in which the user attempted to remove some instrument in the mixture, by vocalizing that part. The results are hard to quantify since these were performed in pre-mixed commercial recordings, but user feedback was in general positive. In most of these cases the user attempted to extract targets such as vocals, lead guitars, and drums[1]. An example set of spectrograms of extracting a vocal track is shown in figure 2. In this particular example the target sound was a female singer, the user was male and spoke the lyrics, as opposed to singing them. Despite these differences the singer's voice was cleanly removed from the recording.

[1] See demo at:`http://www.media.mit.edu/~paris/w9.mov`

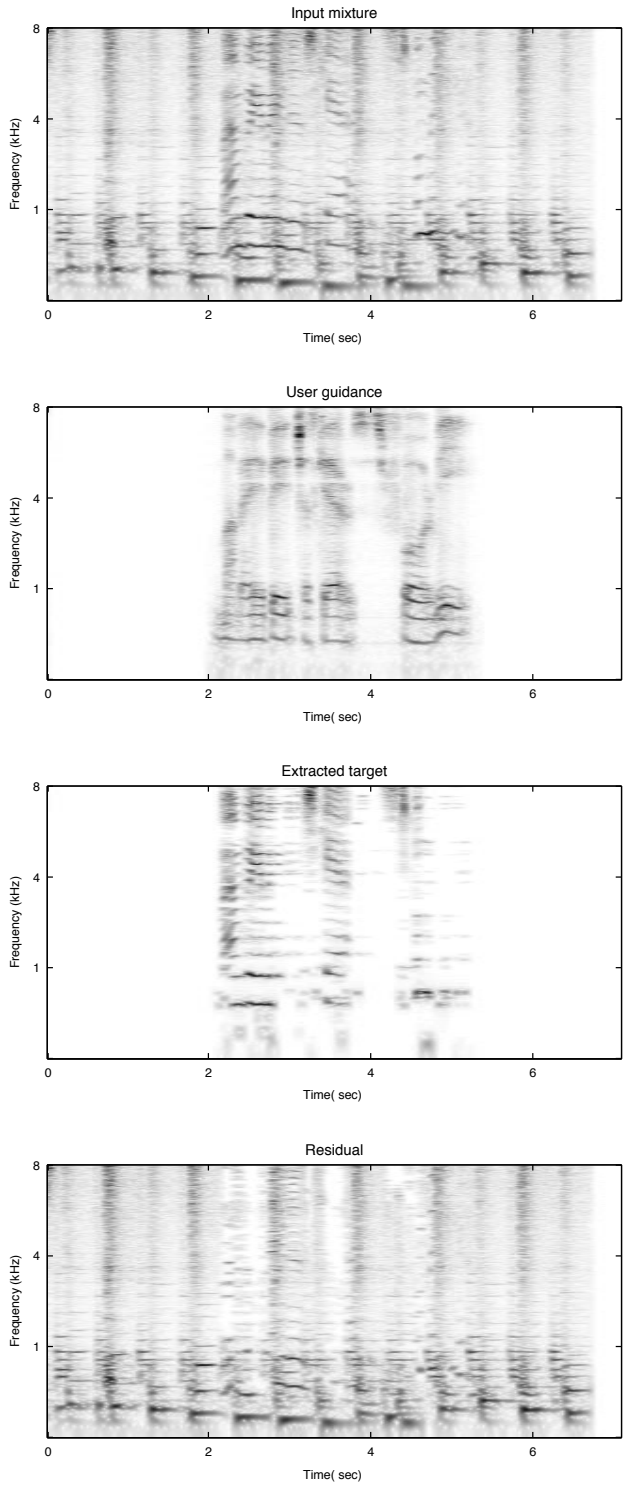

Figure 2: Example results from analysis of "My baby just cares for me" by Nina Simone. The top plot shows a spectrogram of the input mixture. The second plot shows the sound provided by a user. The user was male and spoke the lyrics, roughly in time with the singing in the mixture. The third plot shows the extracted vocal part, and the fourth plot the remainder from the original input.

5. CONCLUSIONS

In this paper we introduced a novel approach for specifying a target sound in a mixture and then extracting it. In it we recruit the user who has to provide a mimicking sound that is then used to help guide a separation algorithm towards a target sound. In contrast to similar work that uses pre-trained target examples, we do not require a close match between the user's input and the target source, but instead use that input as a rough approximation which is then automatically refined. We showed how this approach can help us efficiently and satisfactorily extract targeted sources from complex mixtures, and be used as a new type of interface for mixture modeling.

6. REFERENCES

[1] Smaragdis, P. 1998. Blind separation of convolved mixtures in the frequency domain, in Neurocomputing,vol. 22, pp. 21-34.

[2] H. Sawada, S. Araki, R. Mukai, S. Makino. 2007. Grouping Separated Frequency Components with Estimating Propagation Model Parameters in Frequency-Domain Blind Source Separation," IEEE Trans. Audio, Speech, and Language Processing, vol.15, no.5, pp.1592-1604.

[3] Yilmaz, O., and S. Rickard, 2004. Blind Separation of Speech Mixtures via Time-Frequency Masking, in IEEE Trans. on Signal Processing, Vol. 52, No. 7, pp.1830-1847.

[4] Roweis, S.T. 2003. Factorial Models and Refiltering for Speech Separation and Denoising, in proceedings of Eurospeech 2003 Geneva, Switzerland.

[5] T. Virtanen, A. Mesaros, M. Ryynänen. 2008. Combining Pitch-Based Inference and Non-Negative Spectrogram Factorization in Separating Vocals from Polyphonic Music, Statistical And Perceptual Audition workshop, Interspeech 2008, Brisbane, Australia.

[6] Smaragdis, P., B. Raj, and M.V. Shashanka. 2007. Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures. In proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation. London, UK.

[7] Virtanen, T., and A. T. Cemgil. 2009. Mixtures of Gamma Priors for Non-Negative Matrix Factorization Based Speech Separation, In proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation. Paraty, Brazil.

[8] Févotte, C., N. Bertin and J.-L. Durrieu. 2009. Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis, in Neural Computation, vol. 21, no 3, Mar. 2009

[9] Févotte, C., R. Gribonval and E. Vincent. 2005. BSS EVAL Toolbox User Guide, IRISA Technical Report 1706, Rennes, France, April 2005.