User Guided Audio Selection from Complex Sound Mixtures

Paris Smaragdis
Adobe Systems Inc.

ABSTRACT
In this paper we present a novel interface for selecting sounds in audio mixtures. Traditional interfaces in audio editors provide a graphical representation of sounds which is either a waveform, or some variation of a time/frequency transform. Although with these representations a user might be able to visually identify elements of sounds in a mixture, they do not facilitate object-specific editing (e.g. selecting only the voice of a singer in a song). This interface uses audio guidance from a user in order to select a target sound within a mixture. The user is asked to vocalize (or otherwise sonically represent) the desired target sound, and an automatic process identifies and isolates the elements of the mixture that best relate to the user's input. This way of pointing to specific parts of an audio stream allows a user to perform audio selections which would have been infeasible otherwise.

ACM Classification: H.5.5 [Multimedia Information Systems]: Sound and Music Computing, H.5.2 [User Interfaces]: Voice I/O

General terms: Algorithms, Human Factors

Keywords: Audio interfaces

INTRODUCTION
With the advent of user-friendly software to manipulate media, consumers today are presented with a wide variety of tools with which to edit content they create, or content they wish to experiment with. This trend has been partially fueled by the incorporation of interfaces which allow users to intuitively interact with media. A user who wishes to remove somebody from a photograph can approximately trace the outline of that person and then automatically delete them. A user who wants to change the color of the sky in a video can scribble over the sky to identify that area and then specify the new color for it. Especially in the imaging world, editing of photographs and videos is by now a commonplace operation; one of practical significance but also of artistic expression.

The same does not hold for audio processing. Editing and manipulating complex audio signals presents a unique challenge to users - one that we don't often encounter in other forms of media. Whereas it is relatively simple for a user to point towards specific objects in images and videos, doing so in an audio track is not straightforward. Commonplace recordings such as music, or home video soundtracks are almost always composed out of superimposed sounds that occur simultaneously. Invariably though, a user is most interested in only editing one sound (e.g. the sneeze during the piano concerto, or just the guitar in a music recording). The fact that all these sounds are intertwined inside one waveform, presents a significant challenge in terms of a user interface, since there is no clear way to select a specific sound.

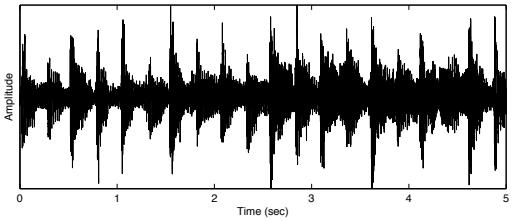

Figure 1: A waveform representation of a piece of music audio. An experienced audio engineer would identify this sound as a piece of music, and would obtain a sense of where the drum beats are, by observing the sharp onsets. However the presence of the singer's voice, or any other instrument is not visually possible to detect. Casual users often use this representation to detect the start and end time of an audio track, but cannot deduce much more information.

This problem has been partially addressed by two distinct fields, the field of audio visualization, and that of sound source separation. In terms of audio visualization, makers of audio processing software have spend significant resources in visualizing audio in forms that help a user understand and manipulate audio. The most widespread (and least informative) representation for audio is the trace of the actual air pressure across time, which is often referred to as the waveform (figure 1). This provides highly accurate visualization of sound, but unfortunately conveys a small amount of information. An experienced user might be able to deduce some basic information using this representation, but in the case of most sound mixtures there is very little information to be found.

In order to present users with more intuitive representations of audio, especially ones that can assist complex editing, audio software is now increasingly relying on *time-frequency* visualizations (often referred to as frequency or spectral representation). Time-frequency decompositions are a family of numerical transforms that allow us to display any time series (like sound) in terms of its time-varying frequency energy content [1]. The most common of these representations is the spectrogram, which one can readily find in many modern audio processing editors. More exotic time-frequency transforms, such as wavelets, warped spectrograms and sinu-

UIST'09,, October 4–7, 2009, Victoria, British Columbia, Canada.

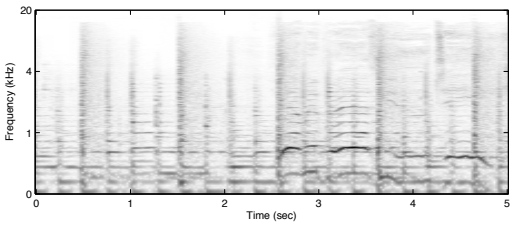

Figure 2: A time-frequency representation of the sound shown in figure 1. An experienced user would be able to visually distinguish some of the constituent sounds. The tall vertical columns represent some of the acoustic energy that belongs to a snare drum. The parallel horizontal wavy tracks in the second half of the recording represent a singers voice. The bass is responsible for most of the pulsating pattern near the bottom of this plot, and the guitar is made up from the shorter vertical columns and the short parallel horizontal lines that stem from them.

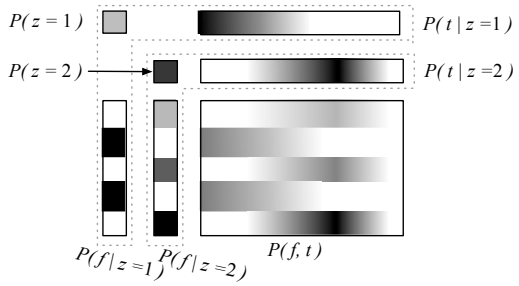

Figure 3: An illustration of a PLCA analysis, on a simplified spectrogram-type input. The input is composed of two patterns, one being two parallel tracks with a sharp onset that subsequently fade out, and the other being three parallel tracks which increase in level and then fade out again. Using a rank-2 decomposition we obtain a succinct analysis of the input. The two distributions $P(f|z)$ describe the two vertical structures we see, their corresponding horizontal distributions $P(t|z)$ show us how each $P(f|z)$ is modulated over time, and $P(z)$ tells us how much these two elements are present overall.

soidal decompositions have also been used, but they effectively communicate the same information to a user. Common to all these visualizations is the ability to show how much acoustic energy exists at a specific point in time and frequency. Since different sounds tend to have different distributions along that space, it is often possible to visually distinguish mixed sounds using such a visualization. See figure 2 for a simple example of interpreting a time-frequency distribution of a musical audio clip. More exotic approaches, such as visualizations of specific parameter estimates from an audio stream (e.g. pitch, or formant structure), are also used on occasion, however these representations are severely limited and do not always produce reliable estimates when analyzing overlapping sounds.

Although any of the representations we mentioned so far can be very informative when one knows what to look for, they still do not facilitate an object-based interaction with audio, such as allowing a user to select and modify a single sound from the mixture. An operation like this would be impossible using the waveform representation due to the fact that each time sample is a sum of multiple sounds. Using a time-frequency representation, we can visualize sound objects, but selecting one of them is a very tedious (if even possible) task for a user. For example selecting the voice in the mixture in figure 2, would involve selecting each of the parallel horizontal lines that are elements of that sound. This presumes that the user can identify which of these lines belong to the voice, and not to the bass of the guitar; a daunting task even for a seasoned expert.

Moving further away from the interface, towards an automated process, we find another approach to manipulate single sounds in a mixture, the approach of automatically separating sounds. The field of sound source separation tackles a very challenging (and still an open) problem, which is that of segmenting a mixture recording in multiple audio tracks that isolate each constituent sound. These approaches work either by taking advantage of spatial statistics in stereo or multichannel recordings, or in the case of single channel recordings, by employing elements of computational psychoacoustics, or by using pre-trained models of sounds that one desires to extract (see [2] for an overview of the state of the art in this field). In most of these cases the state of the art is still in early stages of acceptable performance, and is usually burdened by the fact that it is very hard to reliably point an algorithm towards the particular sound one is interested in. Although in principle these approaches can be a great solution in assisting a user to interact with audio streams, given their present limitations they are more likely to complicate a sound selection interface, rather than make it seamless.

In this paper we will present a new way to interact with sound mixtures, one that is not based on a graphical interface, or automated preprocessing, but one that allows a user to specify a sound using audio guidance. We present a process which can identify which elements of an audio mixture correspond to a sound that the user provides, and then use this as a means of selecting specific sounds from a mixture. This allows the user to use a very intuitive way to specify sounds, which can be as simple as vocalizing (whistling, or otherwise mimicking) the sound that he/she desires to select. In the following sections we present the details of the underlying process, and results that validate its usefulness in this particular task.

SPECTRAL MODELS OF SOUNDS

In this section we will describe the basic model that we use to represent sounds. We start with the introduction of a generic model, and then subsequently add an extension of priors that we can then use for the proposed interface.

Basic Model

The basic model we will use is the Probabilistic Latent Component Analysis (PLCA) based on the work in [3]. This is a simple spectral dictionary model which has been used for various tasks involving mixture sounds. Intuitively this model operates on the a spectrogram representation (such as the one in figure 2), and learns an additive set of basis functions that represent all the potential spectral profiles one expects to see from a sound. More specifically, assuming a time-frequency magnitude distribution $P(f,t)$, we decompose it using:

$$P(f,t) \approx \sum_z P(z)P(f|z)P(t|z). \qquad (1)$$

The model parameters $P(f|z)$, $P(t|z)$ and $P(z)$, can be interpreted as spectral bases, their temporal weights, and basis priors respectively. All of these are indexed by a latent variable z. More simply, $P(f|z)$ defines elements we expect to see in the vertical structure of the input (such as spectra), $P(t|z)$ is their corresponding amount of presence at each time spot, and $P(z)$ is their overall contribution (i.e. how much presence of each spectral basis do we get for each value of z). An illustration of this model is shown in figure 3.

Estimating this model is straightforward using the Expectation-Maximization algorithm, and results in the following iterative estimation equations:

$$\begin{aligned} P(z|f,t) &= \frac{P(z)P(f|z)P(t|z)}{\sum_z P(z)P(f|z)P(t|z)} \\ P(f|z) &= \frac{\sum_t P(f,t)P(z|f,t)}{\sum_{f,t} P(f,t)P(z|f,t)} \\ P(t|z) &= \frac{\sum_f P(f,t)P(z|f,t)}{\sum_{f,t} P(f,t)P(z|f,t)} \\ P(z) &= \frac{\sum_{f,t} P(f,t)P(z|f,t)}{\sum_{z,f,t} P(f,t)P(z|f,t)}. \end{aligned}$$

As shown in [3], one can use this process to learn models of individual sounds, and then use these models to extract such sounds from a mixture. This process involves learning a large number of $P(f|z)$ distributions from recordings of isolated sound sources, and then trying to fit these distributions to an input mixture (i.e. estimating only $P(t|z)$ and $P(z)$). Upon completion of the fitting process one can freely reconstruct the parts of the input that correspond to a particular source, by using only the distributions with the value of z that correspond to the desired source.

Models with Priors

Unfortunately, the model shown in the previous section is not of direct use for the problem we examine here. In order to make it applicable we will introduce a priors extension which will allow us to point towards a source with a high precision, and also remove the requirement of performing an offline estimation on a training data set.

The distributions $P(f|z)$ and $P(t|z)$ that we estimate in the PLCA model are multinomial distributions. In this case, as shown in [4] in a different but quite similar statistical model, the Dirichlet distribution makes for an appropriate prior distribution. The resulting estimation equations for $P(f|z)$ and $P(t|z)$ are changed only slightly by including a term that blends the current estimate with the prior distribution. So for $P(f|z)$ we get:

$$P(f|z) = \frac{\sum_t P(f,t)P(z|f,t) + \kappa\alpha(f|z)}{\sum_{f,t} P(f,t)P(z|f,t) + \kappa\alpha(f|z)}, \qquad (2)$$

and a similar expression for $P(t|z)$. The distribution α is the prior distribution, and the parameter κ decides how much we want to impose the prior in the estimation process. When $\kappa = 0$ we effectively revert to the equation of the basic model shown in the previous sections. This prior effectively biases the results of the estimation of $P(f|z)$ and $P(t|z)$ so that they tend to look more like $\alpha(f|z)$ and $\alpha(t|z)$. We will use this property to employ the PLCA model with priors, for the purposes of this paper.

MATCHING A USER INPUT TO A SOUND IN A MIXTURE

Now let us use the model we developed in the previous sections in order to help a user match a sound in a mixture. Consider once again the mixture time-frequency distribution in figure 2. We will refer to it as the mixture input $P_m(f,t)$. We know that upon analysis with the PLCA model some of the resulting $P(f|z)$ and $P(t|z)$ distributions will correspond to the source we are interested in, whereas the rest will describe the remainder of the input mixture. As it is, there is no way to know which components will correspond to the sound we wish to select. We will therefore require that the user provide a sound (whose time-frequency distribution we denote as $P_u(f,t)$), that is similar to the target and will used as a reference. This does not have to be an exact replication, but rather a rough approximation. An example might be the user singing or whistling the desired part. We will analyze $P_u(f,t)$ with a PLCA model and from it extract a set of spectral components $P_u(f|z)$ and their corresponding temporal weights $P_u(t|z)$. We will then analyze the mixture $P_m(f,t)$ using a PLCA with priors model, this time asking for more components than we used to analyze the user provided sound. We will use the already learned components from the user provided sound ($P_u(f|z)$ and $P_u(t|z)$) as priors for an equal number of components of the mixture, and we will estimate the rest of the mixture components without priors. This process ensures that the components with the priors will latch on to a sound that is similar in spectral and temporal characteristics to the user-provided sound, whereas the rest of the components will explain the remaining sounds. We will start with a large bias parameter κ, and then as the estimated distributions converge we can slowly relax that parameter towards zero. The entire process can run faster than real-time and is briefly summarized as:

- Compute time-frequency distributions $P_u(f,t)$ from the user-provided input, and $P_m(f,t)$ from the mixture sound.
- Obtain N components, $P_u(f|z)$ and $P_u(t|z)$, from $P_u(f,t)$ using PLCA.
- Obtain $N+M$ components from $P_m(f,t)$
  - First N components use $P_u(f|z)$ and $P_u(t|z)$ as priors, with a κ relaxing towards zero during later iterations
  - Remaining components are trained without priors

At the end of this process we can compute the contribution of the desired source sound in the time-frequency space by resynthesizing using only the prior-biased components, whereas the rest of the components will describe the remaining sounds:

$$\begin{aligned} P_{desired}(f,t) &\approx \sum_{z=1}^{N} P(z)P(f|z)P(t|z) \\ P_{remainder}(f,t) &\approx \sum_{z=N+1}^{N+M} P(z)P(f|z)P(t|z) \end{aligned}$$

After this stage the user is free to resynthesize these sources by inverting the time-frequency decomposition as shown in [3], or otherwise process only the time-frequency elements he/she desires. In figure 4 we show an example of this process operating on the mixture shown in figure 2. The user provided a singing of the lead part, which was then used to select only the time-frequency elements that correspond to the singer's part in the mixture.

EXPERIMENTAL VALIDATION

This particular method enables a user to perform an action which would otherwise be impossible to achieve using graphical user interfaces. The two elements we need to consider in evaluating this process as a viable interface, is how accurate it in isolating elements of the desired sound, and

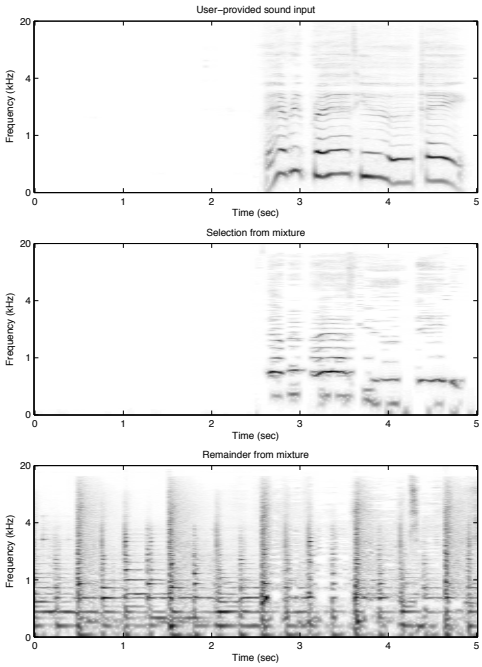

Figure 4: Results from selecting the vocal track from figure 2. The top figure displays the time-frequency distribution of the user input, the middle figure plots the resulting selection given the user guidance, and the third figure plots what remains from the input once the selection is extracted.

what are the tolerances in variance when it comes to the user-provided input. In order to evaluate these two parameters we run three sets of experiments.

For the first experiment we artificially mix speech signals with background music and use the original speech signals as a guide for selecting the speech inside the mixture. This kind of test is known as an oracle case and creates a best case scenario which can reveal the upper performance limits of this approach. For the second experiment we used a similar methodology, but instead of providing the actual speech signal in the recording, that particular speech was uttered by a different speaker of the same gender, attempting to match the speech in the recording. In the third experiment we use a speaker of different gender from the one in the recording. In order to quantify the quality of the sound selection, we use three performance measures, the Signal to Interference Ratio (SIR), the Signal to Distortion Ratio (SDR) and the Signal to Artifacts Ratio (SAR) as proposed in [5]. We averaged these measures from 50 runs of each experiment. The results are shown in the following table:

Metric	Oracle Case	Same Gender	Different Gender
SIR	36.2 dB	13.4 dB	8.3 dB
SDR	11.1 dB	2.7 dB	1.3 dB
SAR	11.2 dB	7.4 dB	3.8 dB

The oracle case results produce audibly perfect selection of the desired sound. The same gender results are also very good in terms of selecting the right sound, but as expected not as accurate as in the oracle case. Finally the different gender case results in slightly worse performance metrics, although it provides a very acceptable degree of distinction between sounds which rivals modern sound separation algorithms. Regardless of the above performance metrics however, the principal application of this approach is to enable a user to select a single sound from a mixture and not to separate it. In most practical scenarios the user will seek to process only one sound out of many (e.g. changing its stereo position, or adding a sound effect) and then keep that sound in the mixture. In such situations it is possible to tolerate comparatively much poorer performance than the one we reported, since processing defects can easily stay undetected by a human listener in the presence of the overall sound mixture.

This technique is not particularly bound to speech signals. We demonstrate its performance on speech due to convenience and easy access to such a corpus, but that approach can be applied to any kind of sound mimicked appropriately. Although it is hard to quantify the performance, using various real-world recordings we have successfully selected guitars using vocalization, solo guitars using whistling, drum patterns using beatboxing, etc. The only requirement is that the user-provided sound correlates stronger with the target than the non-targets in either spectral content or temporal activity. This means that selecting one violin from an orchestra playing in unison is not possible, but it still offers a reasonable solution to a large number of situations with multiple uncorrelated sounds.

CONCLUSIONS

In this paper we presented a new approach to selecting specific sounds from recordings of dense mixtures, a process that is well accepted as impossible to perform with the current state of the art in audio editing software. The user is asked to produce a sound that relates to the source they wish to select, and an automatic process matches the user's input to the most appropriate sound in the mixture. We have shown that the computational implementation that we employ, when provided with a reasonably appropriate user input, can perform a very good job in isolating specific sounds. This validates that this computational core is a promising tool to employ in designing audio driven interfaces for audio editors.

REFERENCES

1. Flandrin, P. 1999. Time-Frequency/Time-scale Analysis, in Wavelet Analysis and Its Applications series, Academic Press; ISBN 978-0-12-259870-8.

2. Makino, S., T-W. Lee, H. Sawada (eds.) 2007. Blind Speech Separation, in Signals and Communication Technology Series, Springer, ISBN: 978-1-4020-6478-4.

3. Smaragdis, P. Raj, B. and Shashanka, M.V. 2007. Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures. In proceedings of ICA2009. London, UK. September 2007.

4. Blei, D., Ng, A., Jordan, M. 2003. Latent Dirichlet allocation. in Journal of Machine Learning Research 3: pp. 9931022. doi:10.1162/jmlr.2003.3.4-5.993.

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