Audio Engineering Society

Convention Paper

Presented at the 122nd Convention
2007 May 5–8 Vienna, Austria

Semi-Automatic Mono to Stereo Up-mixing using Sound Source Formation

Mathieu Lagrange [1], Luis Gustavo Martins [2], and George Tzanetakis [1]

[1] *Computer Science Dept., University of Victoria, BC, Canada*

[2] *INESC Porto, Porto, Portugal*

Correspondence should be addressed to Mathieu Lagrange (lagrange@uvic.ca)

ABSTRACT

In this paper, we propose an original method to include spatial panning information when converting monophonic recordings to stereophonic ones. Sound sources are first identified using perceptually motivated clustering of spectral components. Correlations between these individual sources are then identified to build a middle level representation of the analysed sound. This allows the user to define panning information for major sound sources thus enhancing the stereophonic immersion quality of the resulting sound.

1. INTRODUCTION

The conversion of monophonic recordings to multichannel ones (*upmixing*) is important for several tasks such as old recordings remastering or DVD production. Some methods for mono-to-stereo upmixing proposed in the 50's consider the whole signal and complementarily comb-filter the mono signal for the two stereo channels. According to Schroeder [1], a pseudo-stereophonic effect can be obtained this way. Since then, a lot of work has been done to improve this effect [2, 3].

More recently, stereo-to-5.1 upmixing methods have been proposed in which selective re-panning of individual sound sources of the original recording is applied. Most of these methods consider as input a stereophonic recording and compute an inter-channel coherence measure to identify sound sources [4, 5]. In their method, the panning coefficients corresponding to the various individual sources are then determined by measuring inter-channel similarity.

Dealing with monophonic recordings is much more difficult, since no localization information is available to distinguish between different sound sources. Monceaux [6] propose to segment soundtracks of

movies into music or voice segments based on machine learning algorithms. The proposed approach allows then to selectively use different spatializer depending on the type of segment. This approach can hardly be applied to music recordings, since the voice and musical instruments play together most of the time.

Alternatively, we propose to consider a sound source formation algorithm to identify which frequency components should be re-panned together to preserve the consistency of individual voices during the up-mixing process. As stated before, no spatial information is available to achieve this task. Therefore we consider continuity in time, harmonic relations in frequency and amplitude similarity of the spectral components to identify sound sources. Using the approach proposed in [7, 8], those similarities are computed for dominant frequency components of a Fourier Transform within a texture window of several frames thus providing both frequency and temporal integration. Spectral clustering techniques are then used to identify sound sources using a combination of these similarity measures.

This leads to an algorithm for sound source formation described in Section 2. The extracted sources can be efficiently resynthetised and panned using a Fourier based approach, described in Section 3. Correlations between automatically detected time-frequency clusters are identified in order to group them into larger formations that likely correspond to sound streams. Two types of correlations are studied. First, we describe how the difference between the frequency ranges of different instruments can be used to segregate between sources. We next propose to use timbral similarities between clusters to achieve this task when instruments have the same frequency range. Using this two methods, longer time integration is achieved. In Section 4, some experiments are presented to show the capabilities of the proposed approaches for the upmixing of old jazz recordings . Using this software, the user can easily select sound sources and decides the panning level to be applied to each source. For moderate levels of panning the artifacts of sound source separation and resynthesis are greatly reduced compared to the artifacts when listening to the extracted sources individually. In the last section, we conclude and discuss issues about the extension of the proposed method to mono-to-5.1 upmixing and stereo-to-5.1 upmixing.

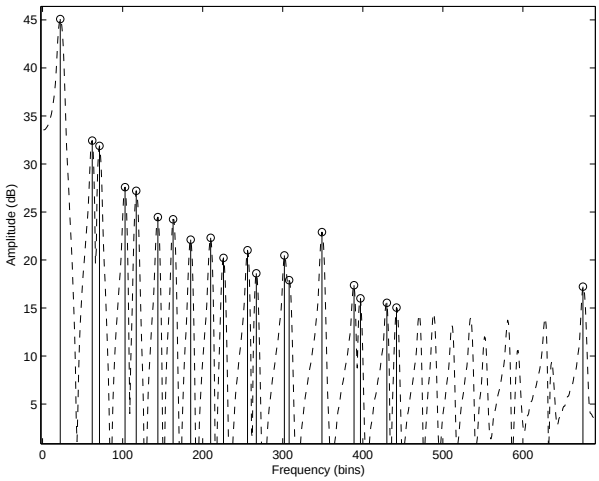

Fig. 1: *Picking of 20 peaks in the spectrum of a mixture of two harmonic sources.*

2. SOUND SOURCE FORMATION

Computational Auditory Scene Analysis (CASA) systems aim at identifying perceived sound sources (e.g. notes in the case of music recordings) and grouping them into auditory streams using psychoacoustical cues [9]. However, as remarked in [10] the precedence rules and the relevance of each of those cues with respect to a given practical task is hard to assess. We intend here to provide a flexible framework where those perceptual cues can be expressed in terms of similarity between time/frequency components. The identification task is then carried out by clustering components which are close in the similarity space. Therefore, the complexity of the algorithm is strongly related to the number of components considered.

2.1. Sinusoidal Modeling

Most CASA approaches consider auditory filterbank or correlogram as a front-end [11]. In this case, the number of time/frequency components is relatively small but closely-spaced components within the same critical band can hardly be separated. Other approaches [12, 13, 10, 14] consider the Fourier Spectrum. In this case, a sufficient frequency resolution is required, which implies a high number of components. Components within the same frequency region can be pre-clustered together according to a

stability criterion of some statistics computed over the considered region. However, this approach has the drawback of introducing another clustering step, and opens the issue of choosing the right descriptors for those pre-clusters. Alternatively, a sinusoidal front-end is helpful to provide meaningful and precise information about the auditory scene while considering only a limited number of components, see Figure 1.

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 stationarity assumption is met. 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 and ϕ_l^k is the phase at the beginning of the frame of the l-th component of L_k sine waves. The f_l and a_l are the frequency and the amplitude of the l-th sine wave, respectively, both of which 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) with a frame size of 46ms and a hop size of 11ms. The precision of these estimates is further improved using phase-based frequency estimators which utilize the relationship between phases of successive frames [15, 16]. Using this enhanced frequency, the rough amplitude estimate provided by the magnitude of the local maximum is also corrected.

2.2. Normalized Cuts

In order to simultaneously optimize partial tracking and source formation, we construct a graph over the entire duration of the sound mixture of interest. Unlike approaches based on local information [17], we utilize the global normalized cut criterion to partition the graph. This criterion as been successfully used for image and video segmentation [18]. In our perspective, each partition is a set of peaks that are grouped together such that the similarity within the partition is minimized and the dissimilarity between different partitions is maximized. The edge weight connecting two peaks p_l^k and $p_{l'}^{k'}$ (k is the frame index and l is the peak index) depends on the proximity of frequency, amplitude and harmonicity:

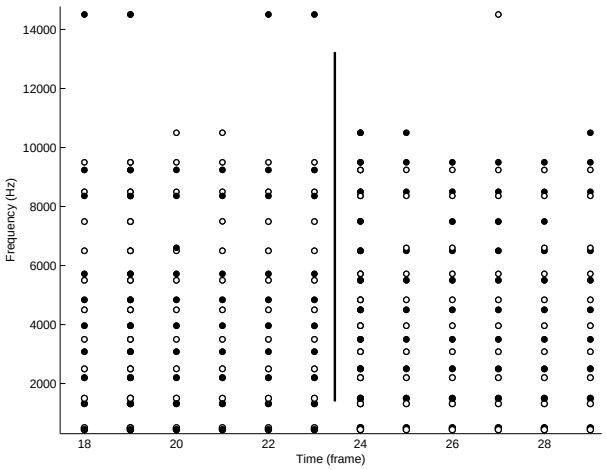

Fig. 2: *Labels assigned by the Ncut algorithm within two different texture windows of six frames before labeling alignment.*

$$W(p_l^k, p_{l'}^{k'}) = W_f(p_l^k, p_{l'}^{k'}) \cdot W_a(p_l^k, p_{l'}^{k'}) \cdot W_h(p_l^k, p_{l'}^{k'})$$

where W_x are typically radial basis functions of distance among the two peaks in the x axis, see [7, 8].

Most existing approaches that apply the Ncut algorithm to audio [12, 14] consider the clustering of components over one analysis frame only. However, the time integration (partial tracking) is as important as the frequency one (source formation) and should be carried out at the same time. We therefore propose in [8] to consider the sinusoidal components extracted within a texture window of several spectral frames (20 in the experiments). Those labels are propagated to the next texture window using an algorithm described in the next section.

2.3. Across-Texture Window Continuity

As discussed in [7], when using the normalized cut approach, it is helpful to implement time continuity between clusters over successive texture windows

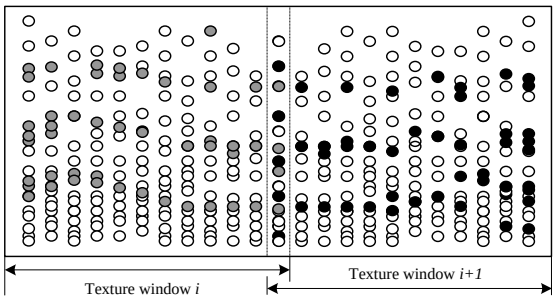

Fig. 3: *The double labeling of the overlapping frame is used to address the over-texture window continuity issue.*

(see Figure 2). Consequently, an overlapping frame is used between each two successive texture windows in order to achieve the desired time continuity (see Figure 3). Within this frame, the peaks are labelled twice; the first labelling comes from the previous texture window and the second is the result of the computation of the Ncut algorithm within the current window. Following [12], we consider maximal intersections between the two labellings in order to derive correspondences between them. Namely, we iteratively look in this intersection frame for the set of peaks with the highest energy, belonging to a cluster of peaks C_i^j in the previous window as well as to another set, C_{i+1}^k, in the current window. Note that those clusters are most likely labelled differently. If those two clusters are judged as compatible, C_{i+1}^k is tagged with the label of C_i^j and those two clusters are discarded. Otherwise only C_i^j, is discarded from the iterative process.

The level of compatibility is computed as the ratio between twice the cumulative amplitude of peaks within the intersection set $C_i^j \cap C_{i+1}^k$ and the cumulative amplitude of peaks within $C_i^j \cup C_{i+1}^k$. Only the peaks of the overlapping frame are considered here. With this algorithm, a sound source that spans several texture windows can be identified as only one cluster of peaks.

3. SOURCE RESYNTHESIS AND PANNING

In order to only resynthesize the selected sources, a bank of sinusoidal oscillators can be used [19]. However, since we intend to pan the selected sources and mix them with the background, this solution can not be utilized due to the phase beating that the mixing may induce. Alternatively, a Fourier based approach is considered. For each analysed frame where a peak of the selected source is present, the corresponding mono signal is windowed using a Hanning window, converted in the polar Fourier domain where the amplitude of each bin is selectively weighted according to the volume and the panning parameters defined by the user (see Figure 4).

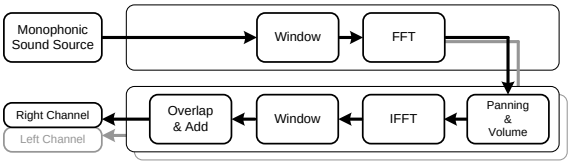

Fig. 4: *Block-diagram of the FFT-based synthesis and panning module.*

For each peak, the frequency region of influence is defined as the FFT bin interval where the peak is dominant. This region is computed as follows. From the closest bin location of the estimated frequency, we look for the closest local magnitude minima in the lower and the higher frequency vicinity. The frequency location of those two minima defines the boundaries of the region of influence. The set of regions corresponding to a cluster forms a mask with the following values:

$$\begin{aligned} m_l(k,t) &= g \cdot (v \cdot (1-p)) + (1-g) m_l(k,t-1) \\ m_r(k,t) &= g \cdot (v \cdot (1+p)) + (1-g) m_r(k,t-1) \end{aligned}$$

if k is inside a region of influence of a peak of a selected cluster. The panning parameter $p \in [-1,1]$, the overall volume v is defined by the user and $g \in [0,1]$ is used for providing smooth changes of the mask over time. We used a $g = 0.8$ in our experiments. Figure 5 provides an example of an instantaneous mask for the first harmonic source extracted from the mixture considered in Figure 2. To achieve the selective synthesis and panning, the magnitude spectrum is weighted by the mask before inverse FFT computation. An example of the evolution along time of the mask for a piano signal is shown in Figure 6.

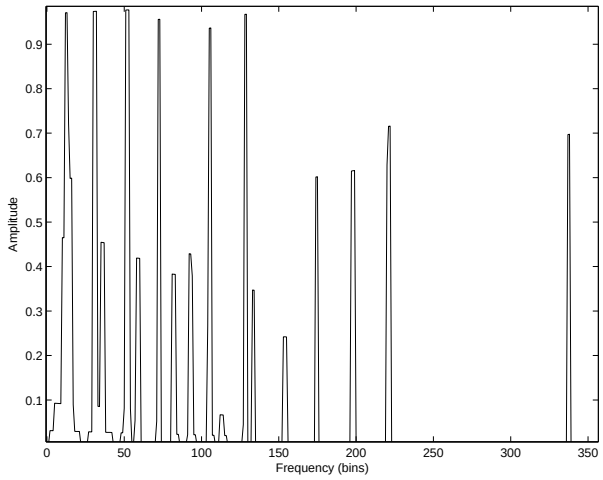

Fig. 5: *Mask of one harmonic source resulting from a mixture of two sources.*

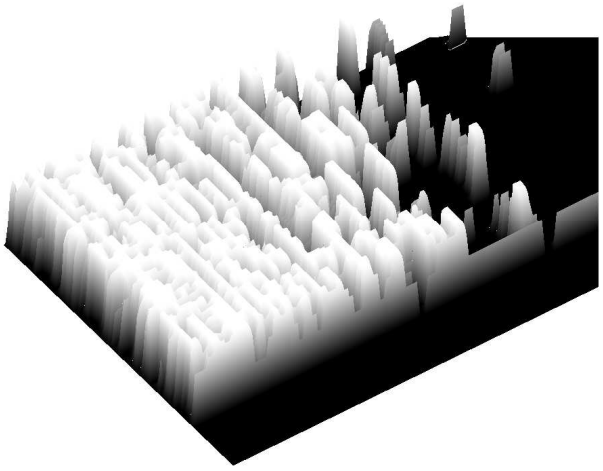

Fig. 6: *Mask of a piano source.*

4. DISCUSSION

This section discusses the use of the approach proposed in this paper for the task of stereo upmixing of old monophonic jazz recordings. For the experiments and evaluation, we made use of the jazz section of the RWC database [20] (mixed down to mono by just using the left channel) and original monophonic recordings of Lester Young and Duke Ellington.

4.1. Resynthesis Quality

Compared to the sinusoidal oscillator approach, the solution based on the inverse FFT allows a better preservation of the brightness of the attack of piano sounds and of the breathiness of instruments like the flute. The low-pass filtering of the mask showed very helpful avoiding magnitude discontinuities between frequency bins of succeeding synthesis frames. However, in complex mixtures, some smearing effects in the high frequency range still prevent the user from using extreme panning values.

4.2. Graphical User Interface Application

Figure 7 depicts the graphic user interface (GUI) developed for a prototype application which allows easy interaction with the system. The GUI presents a spectrogram plot of the audio signal under analysis in the upper part of the window. The lower plot presents the sound sources detected by the sound formation algorithm. Each sound source is represented as a coloured cross, which spans both in time and in frequency as an indication of the sound start time and duration as well as its frequency width. This representation provides a convenient way for visualizing and modifying the auditory scene.

The user can easily select a sound event by clicking its corresponding cross, which will highlight the sound's partials in the spectrogram representation and display additional information, such as start time, duration, mean frequency, mean amplitude and harmonicity level, among others. The user can control both the volume and the panning of the selected sound source, used for the resynthesis of the stereo up-mix. This application was implemented using *Marsyas*, a cross-platform, open source C++ software framework for audio analysis and synthesis [1].

[1] http://marsyas.sourceforge.net

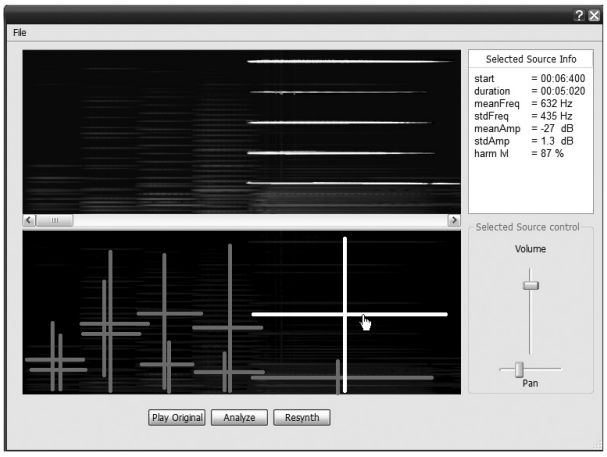

Fig. 7: *Tentative display of the formed clusters. The size and location of the crosses express the main time and frequency features of the clusters.*

4.3. Streams Formation

Any insights regarding similarity between clusters, probably belonging to a same sound stream, are most relevant for the task of semi-automatic panning. However, the formation of streams, namely the clustering of sound sources that have been produced by the same instrument, is a difficult task [11]. We propose two ways to approach this problem. We start by setting constraints on the frequency range of the clusters so it becomes possible to isolate specific instruments, such as the bass line in a polyphonic music piece. Note that contrary to a filtering approach, the frequency range of the segregated streams can overlap significantly without any loss of separability, as depicted in Figure 8.

In case of strongly overlapping frequency ranges, a timbral-based clustering is used. A timbral template is then assigned to each cluster, being defined as the weighted histogram of the frequencies of the peaks within that cluster. The weight is defined as the amplitude of the peak. Once one cluster is selected by the user, so will be all the other clusters with a similar timbral template. This similarity is defined as the cross-correlation between the two templates and thresholded using a user defined parameter.

Our experiments demonstrated that the template derived from a single source tends to be quite specific and consequently has a propensity to identify clusters in the same pitch range. An interesting approach to circumvent this overfitting would be to iteratively update the timbral profile as we group clusters: the algorithm would start with an initial set of clusters defined by the user and then look for highly similar ones. This new group of clusters could now be merged to obtain a more generic template. This template would be more relevant as a timbral descriptor of the considered instrument. We look forward to further investigate towards this direction, planning to include prior knowledge about timbral properties of instruments in the computation of cluster similarity (or even peak similarity - see Equation 2).

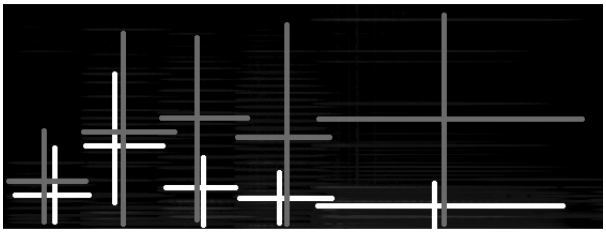

Fig. 8: *Frequency-based stream formation.*

5. CONCLUSION

In this paper we propose a new solution for the semi-automatic mono to stereo panning. The presented technique takes advantage of a new sound source formation approach that is flexible and requires few prior knowledge about the sound sources present in the signal. As long as a source is not hard panned to the left or to the right channels (which would correspond to a sound source separation task), the resulting artifacts will still allow a good quality mono to stereo upmixing.

We plan to generalize this approach to the task of upmixing from stereo to 5.1 recordings, where the spatial location cues exploited in [4, 5] would allow defining an additional similarity cue useful for a better segregation of sources located in the same frequency range.

6. REFERENCES

[1] M. Schroeder, "An artificial stereophonic effect obtained from a single audio signal," *Journal of*

the Audio Engineering Society, vol. 6, no. 2, pp. 74–80, 1958.

[2] M. A. Gerzon, "Signal processing for simulating realistic stereo images," in *93th Convention of the Audio Engineering Society*, 1992.

[3] R. Orban, "A rational technique for synthesizing pseudo-stereo from monophonic sources," *Journal of the Audio Engineering Society*, vol. 18, pp. 157–165, 1970.

[4] C. Avendano, "Frequency domain techniques for stereo to multichannel upmix," in *22th AES International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

[5] ——, "Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppression and re-panning applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

[6] J. Monceaux, F. Pachet, F. Amadu, P. Roy, and A. Zils, "Descriptor-based spatialization," in *118th Convention of the Audio Engineering Society*, 2005.

[7] 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.

[8] M. Lagrange, L. G. Martins, J. Murdoch, and G. Tzanetakis, "Normalized cuts for singing voice separation and melody extraction," *submitted to the IEEE Trans. on Acoustics, Speech, and Signal Processing (Special Issue on Music Information Retrieval)*.

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

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

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

[12] S. Srinivasan and M. Kankanhalli, "Harmonicity and dynamics based audio separation," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 5, 2003, pp. v–640 – v–643.

[13] S. Srinivasan, "Auditory blobs," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). 2004 IEEE International Conference on*, vol. 4, 2004, pp. iv–313 – iv–316.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] L. Girin, S. Marchand, J. di Martino, A. Rbel, and G. Peeters, "Conparing the order of a Polynomial Phase Model for the Synthesis of Quasi-Harmonic Audio Signals," in *WASPAA*. New Paltz, NY, USA: IEEE, October 2003.

[20] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical, and jazz music databases," in *Proc. of Int.Conf. on Music Information Retrieval (ISMIR)*, 2002.