Interactive Music with Active Audio CDs

Sylvain Marchand, Boris Mansencal, and Laurent Girin

LaBRI – CNRS, University of Bordeaux, France
{sylvain.marchand,boris.mansencal}@labri.fr
GIPSA-lab – CNRS, Grenoble Institute of Technology, France
laurent.girin@gipsa-lab.grenoble-inp.fr

Abstract. With a standard compact disc (CD) audio player, the only possibility for the user is to listen to the recorded track, passively: the interaction is limited to changing the global volume or the track. Imagine now that the listener can turn into a musician, playing with the sound sources present in the stereo mix, changing their respective volumes and locations in space. For example, a given instrument or voice can be either muted, amplified, or more generally moved in the acoustic space. This will be a kind of generalized karaoke, useful for disc jockeys and also for music pedagogy (when practicing an instrument). Our system shows that this dream has come true, with active CDs fully backward compatible while enabling interactive music. The magic is that "the music is in the sound": the structure of the mix is embedded in the sound signal itself, using audio watermarking techniques, and the embedded information is exploited by the player to perform the separation of the sources (patent pending) used in turn by a spatializer.

Key words: interactive music, compact disc, audio watermarking, source separation, sound spatialization

1 Introduction

Composers of acousmatic music conduct different stages through the composition process, from sound recording (generally stereophonic) to diffusion (multiphonic). During live interpretation, they interfere decisively on spatialization and coloration of pre-recorded sonorities. For this purpose, the musicians generally use a(n un)mixing console. With two hands, this requires some skill and becomes hardly tractable with many sources or speakers.

Nowadays, the public is also eager to interact with the musical sound. Indeed, more and more commercial CDs come with several versions of the same musical piece. Some are instrumental versions (for karaoke), other are remixes. The karaoke phenomenon gets generalized from voice to instruments, in musical video games such as *Rock Band*[1]. But in this case, to get the interaction the user has to buy the video game, which includes the multitrack recording.

Yet, the music industry is still reluctant to release the multitrack version of musical hits. The only thing the user can get is a standard CD, thus a stereo

[1] see URL: http://www.rockband.com

mix, or its dematerialized version available for download. The CD is not dead: imagine a CD fully backward compatible while permitting musical interaction...

We present the proof of concept of the active audio CD, as a player that can read any active disc – in fact any 16-bit PCM stereo sound file, decode the musical structure present in the sound signal, and use it to perform high-quality source separation. Then, the listener can see and manipulate the sound sources in the acoustic space. Our system is composed of two parts.

First, a CD reader extracts the audio data of the stereo track and decodes the musical structure embedded in the audio signal (Section 2). This additional information consists of the combination of active sources for each time-frequency atom. As shown in [1], this permits an informed source separation of high quality (patent pending). In our current system, we get up to 5 individual tracks out of the stereo mix.

Second, a sound spatializer is able to map in real time all the sound sources to any position in the acoustic space (Section 3). Our system supports either binaural (headphones) or multi-loudspeaker configurations. As shown in [2], the spatialization is done in the spectral domain, is based on acoustics and interaural cues, and the listener can control the distance and the azimuth of each source.

Finally, the corresponding software implementation is described in Section 4.

2 Source Separation

In this section, we present a general overview of the informed source separation technique which is at the heart of the active CD player. This technique is based on a two-step coder-decoder configuration [1][3], as illustrated on Fig. 1. The decoder is the active CD player, that can process separation only on mix signals that have been generated by the coder. At the coder, the mix signal is generated as a linear instantaneous stationary stereo (LISS) mixture, i.e. summation of source signals with constant-gain panning coefficients. Then, the system looks for the two sources that better "explain" the mixture (i.e. the two source signals that are predominant in the mix signal) at different time intervals and frequency channels, and the corresponding source indexes are embedded into the mixture signal as side-information using watermarking. The watermarked mix signal is then quantized to 16-bits PCM. At the decoder, the only available signal is the watermarked and quantized mix signal. The side-information is extracted from the mix signal and used to separate the source signals by a local time / frequency mixture inversion process.

2.1 Time-frequency Decomposition

The voice / instrument source signals are non-stationary, with possibly large temporal and spectral variability, and they generally strongly overlap in the time domain. Decomposing the signals in the time-frequency (TF) domain leads to a sparse representation, i.e. few TF coefficients have a high energy and the overlapping of signals is much lower in the TF domain than in the time domain

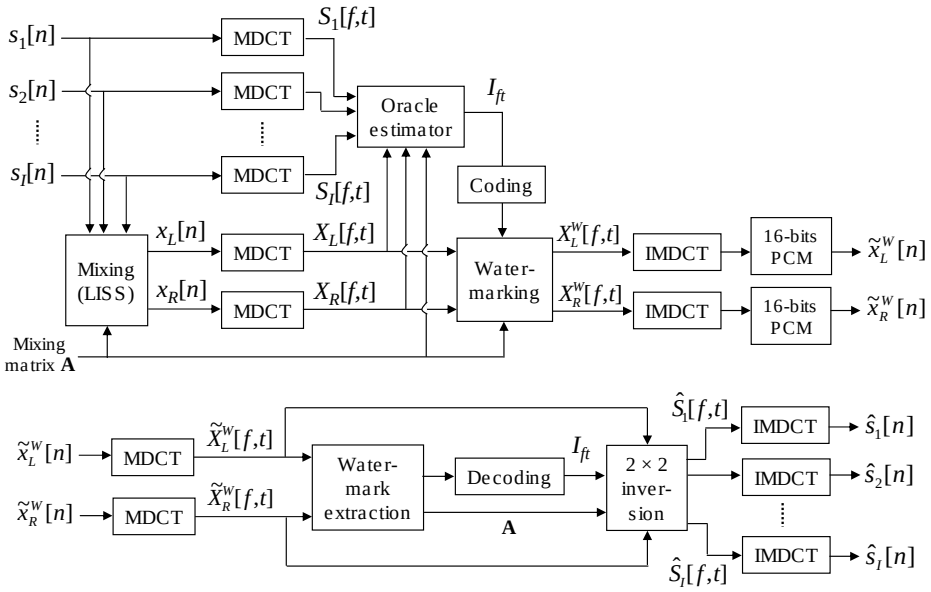

Fig. 1. Informed source separation coder and decoder.

[4][5][6][7]. Therefore, the separation of source signals can be carried out more efficiently in the TF domain. The Modified Discrete Cosine Transform (MDCT) [8] is used as the TF decomposition since it presents several properties very suitable for the present problem: good energy concentration (hence emphasizing audio signals sparsity), very good robustness to quantization (hence robustness to quantization-based watermarking), orthogonality and perfect reconstruction.

Detailed description of the MDCT equations are not provided in the present paper, since it can be found in many papers, e.g. [8]. The MDCT is applied on the source signals and on the mixture signal at the input of the coder to enable the selection of predominant sources in the TF domain. Watermarking of the resulting side-information is applied on the MDCT coefficients of the mix signal and the time samples of the watermarked mix signal are provided by inverse MDCT (IMDCT). At the decoder, the (PCM-quantized) mix signal is MDCT-transformed and the side-information is extracted from the resulting coefficients. Source separation is also carried out in the MDCT domain, and the resulting separated MDCT coefficients are used to reconstruct the corresponding time-domain separated source signals by IMDCT. Technically, the MDCT / IMDCT is applied on signal time frames of $W = 2048$ samples (46.5ms for a sampling frequency $f_s = 44.1$kHz), with a 50%-overlap between consecutive frames (of 1024 frequency bins). The frame length W is chosen to follow the dynamics of music signals while providing a frequency resolution suitable for the separation.

Appropriate windowing is applied at both analysis and synthesis to ensure the "perfect reconstruction" property [8].

2.2 Informed Source Separation

Since the MDCT is a linear transform, the LISS source separation problem remains LISS in the transformed domain. For each frequency bin f and time bin t, we thus have:

$$\mathbf{X}(f,t) = \mathbf{A} \cdot \mathbf{S}(f,t) \tag{1}$$

where $\mathbf{X}(f,t) = [X_1(f,t), X_2(f,t)]^{\mathrm{T}}$ denotes the stereo mixture coefficients vector and $\mathbf{S}(f,t) = [S_1(f,t), \cdots, S_N(f,t)]^{\mathrm{T}}$ denotes the N-source coefficients vector. Because of audio signal sparsity in the TF domain, only at most 2 sources are assumed to be relevant, i.e. of significant energy, at each TF bin (f,t). Therefore, the mixture is locally given by:

$$\mathbf{X}(f,t) \approx \mathbf{A}_{\mathcal{I}_{ft}} \mathbf{S}_{\mathcal{I}_{ft}}(f,t) \tag{2}$$

where $\mathcal{I}_{ft}$ denotes the set of 2 relevant sources at TF bin (f,t). $\mathbf{A}_{\mathcal{I}_{ft}}$ represents the 2×2 mixing sub-matrix made with the $\mathbf{A}_i$ columns of $\mathbf{A}$, $i \in \mathcal{I}_{ft}$. If $\overline{\mathcal{I}}_{ft}$ denotes the complementary set of non-active (or at least poorly active) sources at TF bin (f,t), the source signals at bin (f,t) are estimated by [5]:

$$\begin{cases} \hat{\mathbf{S}}_{\mathcal{I}_{ft}}(f,t) = \mathbf{A}^{-1}_{\mathcal{I}_{ft}} \mathbf{X}(f,t) \\ \hat{\mathbf{S}}_{\overline{\mathcal{I}}_{ft}}(f,t) = 0 \end{cases} \tag{3}$$

where $\mathbf{A}^{-1}_{\mathcal{I}_{ft}}$ denotes the inverse of $\mathbf{A}_{\mathcal{I}_{ft}}$. Note that such a separation technique exploits the 2-channel spatial information of the mixture signal and relaxes the restrictive assumption of a single active source at each TF bin, as made in [4][9][10].

The side-information that is transmitted between coder and decoder (in addition to the mix signal) mainly consists of the coefficients of the mixing matrix $\mathbf{A}$ and the combination of indexes $\mathcal{I}_{ft}$ that identifies the predominant sources in each TF bin. This contrasts with classic blind and semi-blind separation methods where those both types of information have to be estimated from the mix signal only, generally in two steps which can both be a very challenging task and source of significant errors.

As for the mixing matrix, the number of coefficients to be transmitted is quite low in the present LISS configuration[2]. Therefore, the transmission cost of $\mathbf{A}$ is negligible compared to the transmission cost of $\mathcal{I}_{ft}$, and it occupies a very small portion of the watermarking capacity.

As for the source indexes, $\mathcal{I}_{ft}$ is determined at the coder for each TF bin using the source signals, the mixture signal, and the mixture matrix $\mathbf{A}$, as the combination that provides the lower mean squared error (MSE) between the

[2] For 5-source signals, if $\mathbf{A}$ is made of normalized column vectors depending on source azimuths, then we have only 5 coefficients.

original source signals and the estimated source signals obtained with Equation (3) (see [3] for details). This process follows the line of oracle estimators, as introduced in [11] for the general purpose of evaluating the performances of source separation algorithms, especially in the case of underdetermined mixtures and sparse separation techniques. Note that because of the orthogonality / perfect reconstruction property of the MDCT, the selection of the optimal source combination can be processed separately at each TF bin, in spite of the overlap-add operation at source signal reconstruction by IMDCT [11]. When the number of sources is reasonable (typically about 5 for a standard western popular music song), $\tilde{\mathcal{I}}_{ft}$ can be found by exhaustive search, since in contrast to the decoding process, the encoding process is done offline and is therefore not subdue to real-time constraints.

It is important to note that at the coder, the optimal combination is determined from the "original" (unwatermarked) mix signal. In contrast, at the decoder, only the watermarked mix signal is available, and the source separation is obtained by applying Equation (3) using the MDCT coefficients of the watermarked (and 16-bit PCM quantized) mix signal $\tilde{\mathbf{X}}^W(f,t)$ instead of the MDCT coefficients of the original mix signal $\mathbf{X}(f,t)$. However, it has been shown in [3] that the influence of the watermarking (and PCM quantization) process on separation performance is negligible. This is because the optimal combination for each TF bin can be coded with a very limited number of bits before being embedded into the mixture signal. For example, for a 5-source mixture, the number of combinations of two sources among five is 10 and a 4-bit fixed-size code is appropriate (although non optimal) for encoding $\mathcal{I}_{ft}$. In practice, the source separation process can be limited to the [0; 16]kHz bandwidth, because the energy of audio signals is generally very low beyond 16kHz. Since the MDCT decomposition provides as many coefficients as time samples, the side-information bit-rate is $4 \times F_s \times 16,000/(F_s/2) = 128$kbits/s ($F_s = 44,1$kHz is the sampling frequency), which can be split in two 64kbits/s streams, one for each of the stereo channels. This is about 1/4 of the maximum capacity of the watermarking process (see below), and for such capacity, the distortion of the MDCT coefficients by the watermarking process is sufficiently low to not corrupt the separation process of Equation (3). In fact, the main source of degradation in the separation process relies in the sparsity assumption, i.e. the fact that "residual" non-predominant, but non-null, sources may interfere as noise in the local inversion process.

Separation performances are described in details in [3] for "real-world" 5-source LISS music mixtures of different musical styles. Basically, source enhancement from input (mix) to output (separated) ranges from 17dB to 25dB depending on sources and mixture, which is remarkable given the difficulty of such underdetermined mixtures. The rejection of competing sources is very efficient and the source signals are clearly isolated, as confirmed by listening tests. Artefacts (musical noise) are present but are quite limited. The quality of the isolated source signals makes them usable for individual manipulation by the spatializer.

2.3 Watermarking Process

The side-information embedding process is derived from the Quantization Index Modulation (QIM) technique of [12], applied to the MDCT coefficients of the mixture signal in combination with the use of a psycho-acoustic model (PAM) for the control of inaudibility. It has been presented in details in [13][14]. Therefore, we just present the general lines of the watermarking process in this section, and we refer the reader to these papers for technical details.

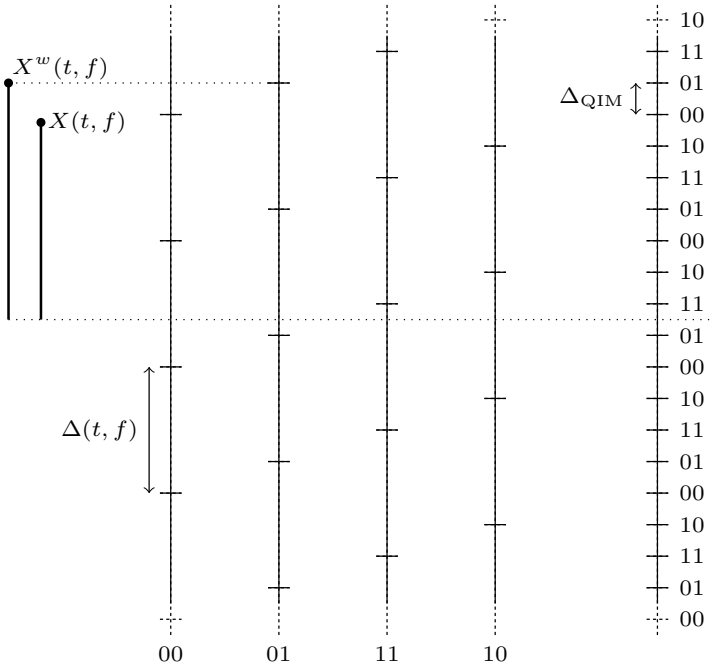

Fig. 2. Example of QIM using a set of quantizers for $C(t,f) = 2$ and the resulting global grid. We have $\Delta(t,f) = 2^{C(t,f)} \cdot \Delta_{\mathrm{QIM}}$. The binary code 01 is embedded into the MDCT coefficient $X(t,f)$ by quantizing it to $X^w(t,f)$ using the quantizer indexed by 01.

The embedding principle is the following. Let us denote by $C(t,f)$ the capacity at TF bin (t,f), i.e. the maximum size of the binary code to be embedded in the MDCT coefficient at that TF bin (under inaudibility constraint). We will see below how $C(t,f)$ is determined for each TF bin. For each TF bin (t,f), a set of $2^{C(t,f)}$ uniform quantizers is defined, which quantization levels are intertwined, and each quantizer represents a $C(t,f)$-bit binary code. Embedding a given binary code on a given MDCT coefficient is done by quantizing this coefficient with the corresponding quantizer (i.e. the quantizer indexed by the code

to transmit; see Fig. 2). At the decoder, recovering the code is done by comparing the transmitted MDCT coefficient (potentially corrupted by transmission noise) with the $2^{C(t,f)}$ quantizers, and selecting the quantizer with the quantization level closest to the transmitted MDCT coefficient. Note that because the capacity values depend on (f,t), those values must be transmitted to the decoder to select the right set of quantizers. For this, a fixed-capacity embedding "reservoir" is allocated in the higher frequency region of the spectrum, and the capacity values are actually defined within subbands (see [14] for details). Note also that the complete binary message to transmit (here the set of $\mathcal{I}_{ft}$ codes) is split and spread across the different MDCT coefficients according to the local capacity values, so that each MDCT coefficient carries a small part of the complete message. Conversely, the decoded elementary messages have to be concatenated to recover the complete message. The embedding rate R is given by the average total number of embedded bits per second of signal. It is obtained by summing the capacity $C(t,f)$ over the embedded region of the TF plane and dividing the result by the signal duration.

The performance of the embedding process is determined by two related constraints: the watermark decoding must be robust to the 16-bit PCM conversion of the mix signal (which is the only source of noise because the "perfect reconstruction" property of MDCT ensures transparency of IMDCT/MDCT chained processes), and the watermark must be inaudible. The time-domain PCM quantization leads to additive white Gaussian noise on MDCT coefficients, which induces a lower bound for Δ_{QIM} the minimum distance between two different levels of all QIM quantizers (see Fig. 2). Given that lower bound, the inaudibility constraint induces an upper bound on the number of quantizers, hence a corresponding upper bound on the capacity $C(t,f)$ [13][14]. More specifically, the constraint is that the power of the embedding error in the worst case remains under the masking threshold $M(t,f)$ provided by a psychoacoustic model. The PAM is inspired from the MPEG-AAC model [15] and adapted to the present data hiding problem. It is shown in [14] that the optimal capacity is given by:

$$C^{\alpha}(t,f) = \left\lfloor \frac{1}{2}\log_2\left(\frac{M(t,f)\cdot 10^{\frac{\alpha}{10}}}{\Delta^2_{\mathrm{QIM}}}\right)+1 \right\rfloor \tag{4}$$

where $\lfloor . \rfloor$ denotes the *floor* function, and α is a scaling factor (in dB) that enables users to control the trade-off between signal degradation and embedding rate by translating the masking threshold. Signal quality is expected to decrease as embedding rate increases, and vice-versa. When $\alpha > 0$dB, the masking threshold is raised. Larger values of the quantization error allow for larger capacities (and thus higher embedding rate), at the price of potentially lower quality. At the opposite, when $\alpha < 0$dB, the masking threshold is lowered, leading to a "safety margin" for the inaudibility of the embedding process, at the price of lower embedding rate. It can be shown that the embedding rate R^{α} corresponding to C^{α} and the basic rate $R = R^0$ are related by [14]:

$$R^{\alpha} \simeq R + \alpha \cdot \frac{\log_2(10)}{10} \cdot F_u \tag{5}$$

(F_u being the bandwidth of the embedded frequency region). This linear relation enables to easily control the embedding rate by the setting of α.

The inaudibility of the watermarking process has been assessed by subjective and objective tests. In [13][14], Objective Difference Grade (ODG) scores [16][17] were calculated for a large range of embedding rates and different musical styles. ODG remained very close to zero (hence imperceptibility of the watermark) for rates up to about 260kbps for musical styles such as pop, rock, jazz, funk, bossa, fusion, etc. (and "only" up to about 175kbps for classical music). Such rates generally correspond to the basic level of the masking curve allowed by the PAM (i.e. $\alpha = 0$dB). More "comfortable" rates can be set between 150 and 200kbits/s to guarantee transparent quality for the embedded signal. This flexibility is used in our informed source separation system to fit the embedding capacity with the bit-rate of the side-information, which is at the very reasonable value of 64kbits/s/channel. Here, the watermarking is guaranteed to be "highly inaudible", since the masking curve is significantly lowered to fit the required capacity.

3 Sound Spatialization

Now that we have recovered the different sound sources present in the original mix, we can allow the user to manipulate them in space. We consider each punctual and omni-directional sound source in the horizontal plane, located by its (ρ, θ) coordinates, where ρ is the distance of the source to the head center and θ is the azimuth angle. Indeed, as a first approximation in most musical situations, both the listeners and instrumentalists are standing on the (same) ground, with no relative elevation. Moreover, we consider that the distance ρ is large enough for the acoustic wave to be regarded as planar when reaching the ears.

3.1 Acoustic Cues

In this section, we intend to perform real-time high-quality (convolutive) mixing. The source s will reach the left (L) and right (R) ears through different acoustic paths, characterizable with a pair of filters, which spectral versions are called Head-Related Transfer Functions (HRTFs). HRTFs are frequency- and subject-dependent. The CIPIC database [18] samples different listeners and directions of arrival.

A sound source positioned to the left will reach the left ear sooner than the right one, in the same manner the right level should be lower due to wave propagation and head shadowing. Thus, the difference in amplitude or Interaural Level Difference (ILD, expressed in decibels – dB) [19] and difference in arrival time or Interaural Time Difference (ITD, expressed in seconds) [20] are the main spatial cues for the human auditory system [21].

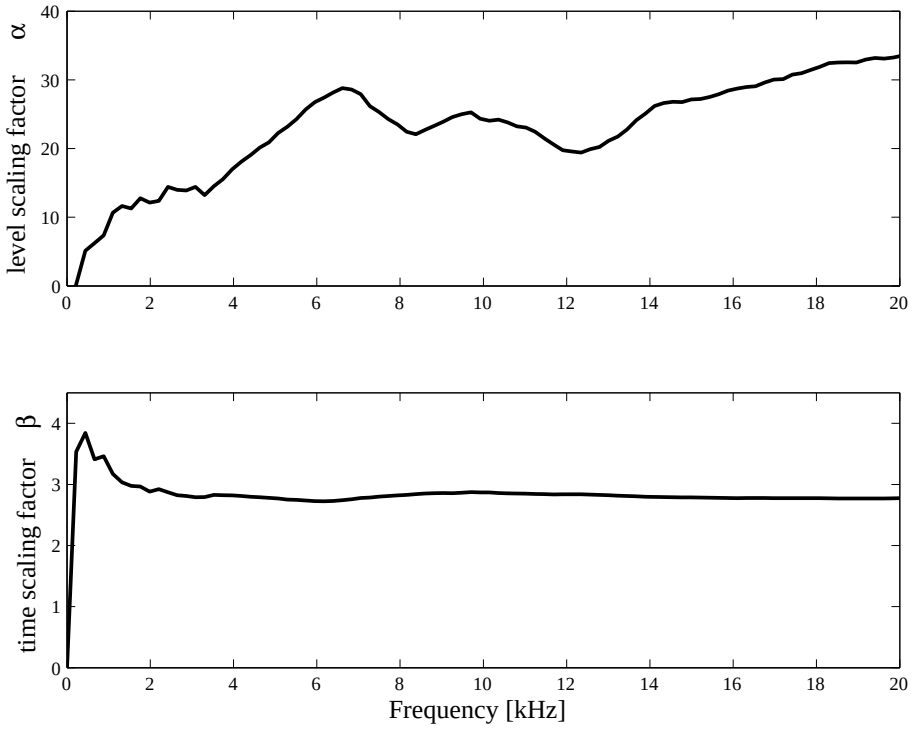

Fig. 3. Frequency-dependent scaling factors: α (top) and β (bottom).

Interaural Level Differences. After Viste [22], the ILDs can be expressed as functions of $\sin(\theta)$, thus leading to a sinusoidal model:

$$\mathrm{ILD}(\theta, f) = \alpha(f)\sin(\theta) \tag{6}$$

where $\alpha(f)$ is the average scaling factor that best suits our model, in the least-square sense, for each listener of the CIPIC database (see Fig. 3). The overall error of this model over the CIPIC database for all subjects, azimuths, and frequencies is of 4.29dB.

Interaural Time Differences. Because of the head shadowing, Viste uses for the ITDs a model based on $\sin(\theta) + \theta$, after Woodworth [23]. However, from the theory of the diffraction of an harmonic plane wave by a sphere (the head), the ITDs should be proportional to $\sin(\theta)$. Contrary to the model by Kuhn [24], our model takes into account the inter-subject variation and the full-frequency band. The ITD model is then expressed as:

$$\mathrm{ITD}(\theta, f) = \beta(f) r \sin(\theta)/c \tag{7}$$

where β is the average scaling factor that best suits our model, in the least-square sense, for each listener of the CIPIC database (see Fig. 3), r denotes the

head radius, and c is the sound celerity. The overall error of this model over the CIPIC database is 0.052ms (thus comparable to the 0.045ms error of the model by Viste).

Distance Cues. In ideal conditions, the intensity of a source is halved (decreases by -6dB) when the distance is doubled, according to the well-known Inverse Square Law [25]. Applying only this frequency-independent rule to a signal has no effect on the sound timbre. But when a source moves far from the listener, the high frequencies are more attenuated than the low frequencies. Thus the sound spectrum changes with the distance. More precisely, the spectral centroid moves towards the low frequencies as the distance increases. In [26], the authors show that the frequency-dependent attenuation due to atmospheric attenuation is roughly proportional to f^2, similarly to the ISO 9613-1 norm [27]. Here, we manipulate the magnitude spectrum to simulate the distance between the source and the listener. Conversely, we would measure the spectral centroid (related to brightness) to estimate the source's distance to listener.

In a concert room, the distance is often simulated by placing the speaker near / away from the auditorium, which is sometimes physically restricted in small rooms. In fact, the architecture of the room plays an important role and can lead to severe modifications in the interpretation of the piece.

Here, simulating the distance is a matter of changing the magnitude of each short-term spectrum X. More precisely, the ISO 9613-1 norm [27] gives the frequency-dependent attenuation factor in dB for given air temperature, humidity, and pressure conditions. At distance ρ, the magnitudes of $X(f)$ should be attenuated by $D(f,\rho)$ decibels:

$$D(f,\rho) = \rho \cdot a(f) \tag{8}$$

where $a(f)$ is the frequency-dependent attenuation, which will have an impact on the brightness of the sound (higher frequencies being more attenuated than lower ones).

More precisely, the total absorption in decibels per meter $a(f)$ is given by a rather complicated formula:

$$\begin{aligned}\frac{a(f)}{P} \approx 8.68 \cdot F^2 \Big\{ & 1.84 \cdot 10^{-11} \left(\tfrac{T}{T_0}\right)^{\frac{1}{2}} P_0 + \left(\tfrac{T}{T_0}\right)^{-\frac{5}{2}} \\ & \Big[0.01275 \cdot e^{-2239.1/T} / [F_{r,O} + (F^2/F_{r,O})] \\ & + 0.1068 \cdot e^{-3352/T} / [F_{r,N} + (F^2/F_{r,N})]\Big]\Big\}\end{aligned} \tag{9}$$

where $F = f/P$, $F_{r,O} = f_{r,O}/P$, $F_{r,N} = f_{r,N}/P$ are frequencies scaled by the atmospheric pressure P, and P_0 is the reference atmospheric pressure (1 atm), f is the frequency in Hz, T is the atmospheric temperature in Kelvin (K), T_0 is the reference atmospheric temperature (293.15K), $f_{r,O}$ is the relaxation frequency of molecular oxygen, and $f_{r,N}$ is the relaxation frequency of molecular nitrogen (see [26] for details).

The spectrum thus becomes:

$$X(\rho, f) = 10^{(X_{dB}(t,f) - D(f,\rho))/20} \tag{10}$$

where X_{dB} is the spectrum X in dB scale.

3.2 Binaural Spatialization

In binaural listening conditions using headphones, the sound from each earphone speaker is heard only by one ear. Thus the encoded spatial cues are not affected by any cross-talk signals between earphone speakers.

To spatialize a sound source to an expected azimuth θ, for each short-term spectrum X, we compute the pair of left (X_L) and right (X_R) spectra from the spatial cues corresponding to θ, using Equations (6) and (7), and:

$$X_L(t,f) = H_L(t,f)X(t,f) \text{ with } H_L(t,f) = 10^{+\Delta_a(f)/2} e^{+j\Delta_\phi(f)/2}, \tag{11}$$
$$X_R(t,f) = H_R(t,f)X(t,f) \text{ with } H_R(t,f) = 10^{-\Delta_a(f)/2} e^{-j\Delta_\phi(f)/2} \tag{12}$$

(because of the symmetry among the left and right ears), where Δ_a and Δ_ϕ are given by:

$$\Delta_a(f) = \text{ILD}(\theta, f)/20, \tag{13}$$
$$\Delta_\phi(f) = \text{ITD}(\theta, f) \cdot 2\pi f. \tag{14}$$

This is indeed a convolutive model, the convolution turning into a multiplication in the spectral domain. Moreover, the spatialization coefficients are complex. The control of both amplitude and phase should provide better audio quality [28] than amplitude-only spatialization. Indeed, we reach a remarkable spatialization realism through informal listening tests with AKG K240 Studio headphones.

3.3 Multi-Loudspeaker Spatialization

In a stereophonic display, the sound from each loudspeaker is heard by both ears. Thus, as in the transaural case, the stereo sound reaches the ears through four acoustic paths, corresponding to transfer functions (C_{ij}, i representing the speaker and j the ear), see Fig. 4. Here, we generate these paths artificially using the binaural model (using the distance and azimuth of the source to the ears for H, and of the speakers to the ears for C). Since we have:

$$X_L = H_L X = C_{LL} \underbrace{K_L X}_{Y_L} + C_{LR} \underbrace{K_R X}_{Y_R} \tag{15}$$
$$X_R = H_R X = C_{RL} \underbrace{K_L X}_{Y_L} + C_{RR} \underbrace{K_R X}_{Y_R} \tag{16}$$

the best panning coefficients under CIPIC conditions for the pair of speakers to match the binaural signals at the ears (see Equations (11) and (12)) are then given by:

$$K_L(t,f) = C \cdot (C_{RR} H_L - C_{LR} H_R)\,, \tag{17}$$
$$K_R(t,f) = C \cdot (-C_{RL} H_L + C_{LL} H_R) \tag{18}$$

with the determinant computed as:

$$C = 1/\left(C_{LL} C_{RR} - C_{RL} C_{LR}\right). \tag{19}$$

During diffusion, the left and right signals (Y_L, Y_R) to feed left and right speakers are obtained by multiplying the short-term spectra X with K_L and K_R, respectively:

$$Y_L(t,f) = K_L(t,f) X(t,f) = C \cdot (C_{RR} X_L - C_{LR} X_R)\,, \tag{20}$$
$$Y_R(t,f) = K_R(t,f) X(t,f) = C \cdot (-C_{RL} X_L + C_{LL} X_R)\,. \tag{21}$$

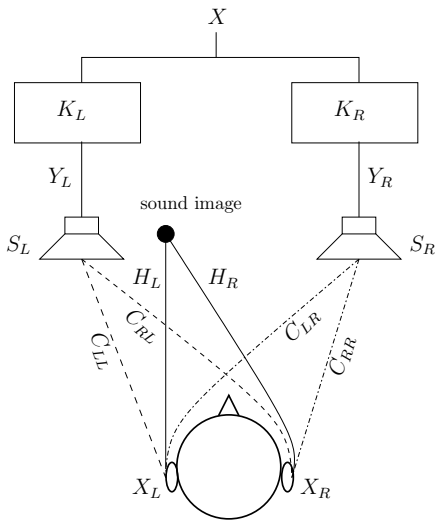

Fig. 4. Stereophonic loudspeaker display: the sound source X reaches the ears L, R through four acoustic paths (C_{LL}, C_{LR}, C_{RL}, C_{RR}).

In a setup with many speakers we use the classic pair-wise paradigm [29], consisting in choosing for a given source only the two speakers closest to it (in azimuth): one at the left of the source, the other at its right (see Fig. 5). The left and right signals computed for the source are then dispatched accordingly.

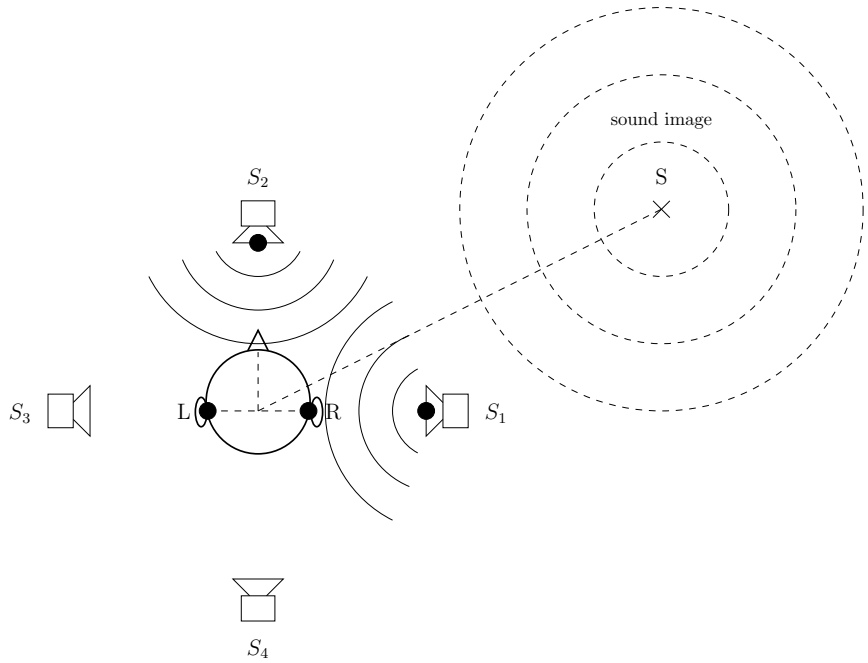

Fig. 5. Pairwise paradigm: for a given sound source, signals are dispatched only to the two speakers closest to it (in azimuth).

4 Software System

Our methods for source separation and sound spatialization have been implemented as a real-time software system, programmed in C++ language and using Qt4[3], JACK[4], and FFTW[5]. These libraries were chosen to ensure portability and performance on multiple platforms. The current implementation has been tested on Linux and MacOS X operating systems, but should work with very minor changes on other platforms, e.g. Windows.

Fig. 6 shows an overview of the architecture of our software system. Source separation and sound spatialization are implemented as two different modules. We rely on JACK audio ports system to route audio streams between these two modules in real time.

This separation in two modules was mainly dictated by a different choice of distribution license: the source separation of the active player should be patented and released without sources, while the spatializer will be freely available under the GNU General Public License.

[3] see URL: `http://trolltech.com/products/qt`
[4] see URL: `http://jackaudio.org`
[5] see URL: `http://www.fftw.org`

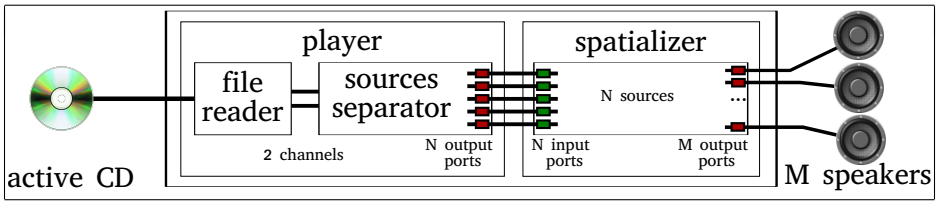

Fig. 6. Overview of the software system architecture.

4.1 Usage

Player. The active player is presented as a simple audio player, based on JACK. The graphical user interface (GUI) is a very common player interface. It allows to play or pause the reading / decoding. The player reads "activated" stereo files, from an audio CD or file, and then decodes the stereo mix in order to extract the N (mono) sources. Then these sources are transferred to N JACK output ports, currently named `QJackPlayerSeparator:output`i, with i in [1; N].

Spatializer. The spatializer is also a real-time application, standalone and based on JACK. It has N inputs ports that correspond to the N sources to spatialize. These ports are to be connected, with the JACK ports connection system, to the N outputs ports of the active player. The spatializer can be configured to work with headphones (binaural configuration) or with M loudspeakers.

Fig. 7 shows the current interface of the spatializer, which displays a bird's eye view of the audio scene. The user's avatar is in the middle, represented by a head viewed from above. He is surrounded by various sources, represented as notes in colored discs. When used in a multi-speaker configuration, speakers may be represented in the scene. If used in a binaural configuration, the user's avatar is represented wearing headphones.

With this graphical user interface, the user can interactively move each source individually. He picks one of the source representation and drags it around. The corresponding audio stream is then spatialized, in real time, according to the new source position (distance and azimuth). The user can also move his avatar among the sources, as if the listener was moving on the stage, between the instrumentalists. In this situation, the spatialization changes for all the sources simultaneously, according to their new relative positions to the moving user avatar.

Inputs and outputs are set via two configuration files (see Fig. 8). A source configuration file defines the number of sources. For each source, this file gives the name of the output port to which a spatializer input port will be connected, and also its original azimuth and distance. Fig. 8 shows the source configuration file to connect to the active player with 5 ports. A speaker configuration file defines the number of speakers. For each speaker, this file gives the name of the physical (soundcard) port to which a spatializer output port will be connected,

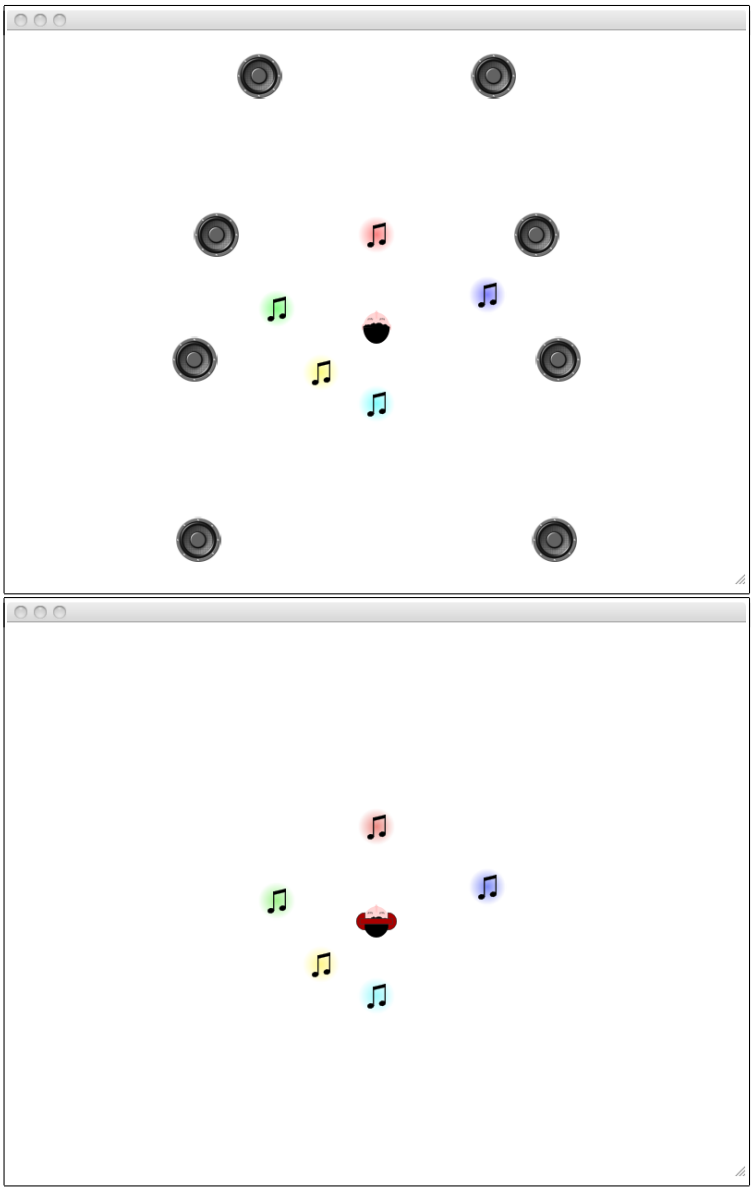

Fig. 7. From the stereo mix stored on the CD, our player is allowing the listener (*center*) to manipulate 5 sources in the acoustic space, using here an octophonic display (top) or headphones (bottom).

and the azimuth and distance of the speaker. The binaural case is distinguished by the fact that it has only two speakers with neither azimuth nor distance specified. Fig. 8 shows the speaker configuration files for binaural and octophonic (8-speaker) configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE source_configuration>
<source_configuration version="1.0">
    <source port="QJackPlayerSeparator:output1" azimuth="-40" distance="1"></source>
    <source port="QJackPlayerSeparator:output2" azimuth="-30" distance="1"></source>
    <source port="QJackPlayerSeparator:output3" azimuth="0"   distance="1"></source>
    <source port="QJackPlayerSeparator:output4" azimuth="20"  distance="1"></source>
    <source port="QJackPlayerSeparator:output5" azimuth="40"  distance="1"></source>
</source_configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE speaker_configuration>
<speaker_configuration version="1.0">
    <speaker port="system:playback_2"></speaker>
    <speaker port="system:playback_1"></speaker>
</speaker_configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE speaker_configuration>
<speaker_configuration version="1.0">
    <speaker port="system:playback_1" azimuth="25" distance="3"></speaker>
    <speaker port="system:playback_2" azimuth="-25" distance="3"></speaker>
    <speaker port="system:playback_3" azimuth="-60" distance="2"></speaker>
    <speaker port="system:playback_4" azimuth="-100" distance="2"></speaker>
    <speaker port="system:playback_5" azimuth="-140" distance="3"></speaker>
    <speaker port="system:playback_6" azimuth="140" distance="3"></speaker>
    <speaker port="system:playback_7" azimuth="100" distance="2"></speaker>
    <speaker port="system:playback_8" azimuth="60" distance="2"></speaker>
</speaker_configuration>
```

Fig. 8. Example of configuration files: 5-source configuration (top), binaural output configuration (middle), and then 8-speaker configuration (bottom) files.

4.2 Implementation

Player. The current implementation is divided into three threads. The main thread is the Qt GUI. A second thread reads and bufferizes data from the stereo file, to be able to compensate for any physical CD reader latency. The third thread is the JACK process function. It separates the data for the N sources and feeds the output ports accordingly. In the current implementation, the number of output sources is fixed to $N = 5$.

Our source separation implementation is rather efficient as for a Modified Discrete Cosine Transform (MDCT) of W samples, we only do a Fast Fourier Transform (FFT) of size $W/4$. Indeed, a MDCT of length W is almost equivalent to a type-IV DCT of length $W/2$ that can be computed with a FFT of length

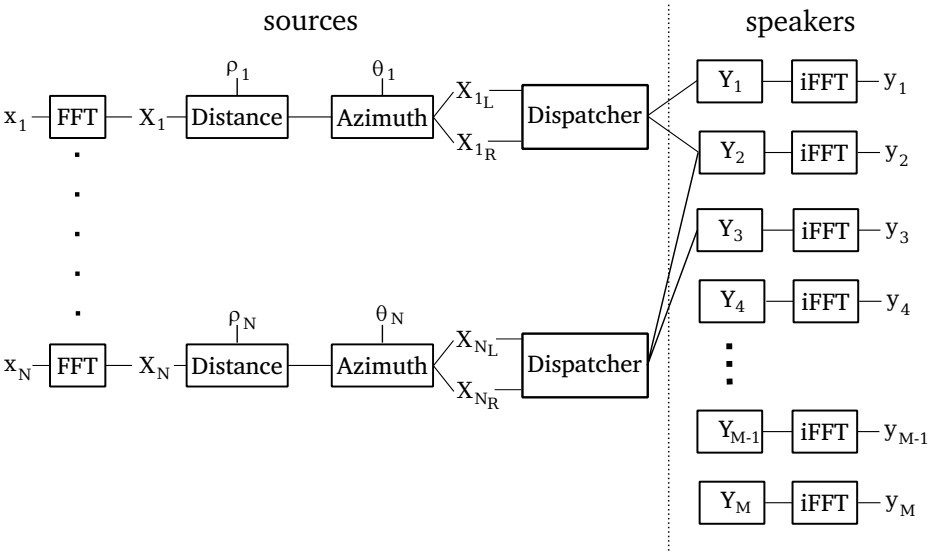

Fig. 9. Processing pipeline for the spatialization of N sources on M speakers.

$W/4$. Thus, as we use MDCT and IMDCT of size $W = 2048$, we only do FFT and IFFT of 512 samples.

Spatializer. The spatializer is currently composed of two threads: a main thread, the Qt GUI, and the JACK process function.

Fig. 9 shows the processing pipeline for the spatialization. For each source, x_i is first transformed into the spectral domain with a FFT to obtain its spectrum X_i. This spectrum is attenuated for distance ρ_i (see Equation (10)). Then, for an azimuth θ_i, we obtain the left (X_{iL}) and right (X_{iR}) spectra (see Equations (11) and (12)). The dispatcher then chooses the pair $(j, j+1)$ of speakers surrounding the azimuth θ_i, transforms the spectra X_{iL} and X_{iR} by the coefficients corresponding to this speaker pair (see Equations (20) and (21)), and adds the resulting spectra Y_j and Y_{j+1} in the spectra of these speakers. Finally, for each speaker, its spectrum is transformed with an IFFT to obtain back in the time domain the mono signal y_j for the corresponding output.

Source spatialization is more computation-intensive than source separation, mainly because it requires more transforms (N FFTs and M IFFTs) of larger size $W = 2048$. For now, source spatialization is implemented as a serial process. However, we can see that this pipeline is highly parallel. Indeed, almost everything operates on separate data. Only the spectra of the speakers may be accessed concurrently, to accumulate the spectra of sources that would be spatialized to the same or neighbouring speaker pairs. These spectra should then be protected with mutual exclusion mechanisms. A future version will take advantage of multi-core processor architectures.

4.3 Experiments

Our current prototype has been tested on an Apple MacBook Pro, with an Intel Core 2 Duo 2.53GHz processor, connected to headphones or to a 8-speaker system, via a MOTU 828 MKII soundcard. For such a configuration, the processing power is well contained. In order to run in real time, given a signal sampling frequency of 44.1kHz and windows of 2048 samples, the overall processing time should be less than 23ms. With our current implementation, 5-source separation and 8-speaker spatialization, this processing time is in fact less than 3ms on the laptop mentioned previously. Therefore, the margin to increase the number of sources to separate and/or the number of loudspeakers is quite confortable. To confirm this, we exploited the split of the source separation and spatialization modules to test the spatializer without the active player, since the latter is currently limited to 5 sources. We connected to the spatializer a multi-track player that reads several files simultaneously and exposes these tracks as JACK output ports. Tests showed that the spatialization can be applied to roughly 48 sources on 8 speakers, or 40 sources on 40 speakers on this computer.

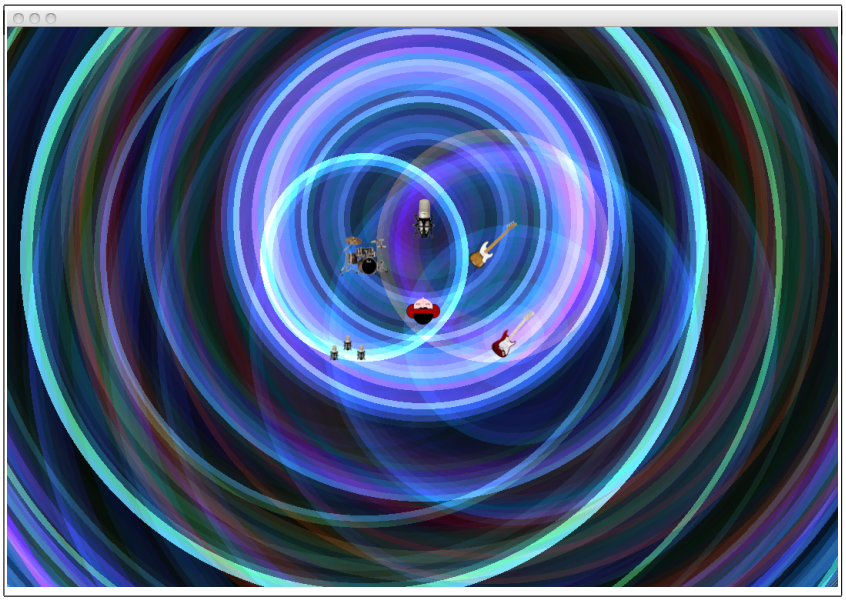

Fig. 10. Enhanced graphical interface with pictures of instruments for sources and propagating sound waves represented as colored circles.

These performances allow us to have some processing power for other computations, to improve user experience for example. Fig. 10 shows an example of an enhanced graphical interface where the sources are represented with pictures of the instruments, and the propagation of the sound waves is represented for

each source by time-evolving colored circles. The color of each circle is computed from the color (spectral envelope) of the spectrum of each source and updated in real time as the sound changes.

5 Conclusion and Future Work

We have presented a real-time system for musical interaction from stereo files, fully backward-compatible with standard audio CDs. This system consists of a source separator and a spatializer.

The source separation is based on the sparsity of the source signals in the spectral domain and the exploitation of the stereophony. This system is characterized by a quite simple separation process and by the fact that some side-information is inaudibly embedded in the signal itself to guide the separation process. Compared to (semi-)blind approaches also based on sparsity and local mixture inversion, the informed aspect of separation guarantees the optimal combination of the sources, thus leading to a remarkable increase of quality of the separated signals.

The sound spatialization is based on a simplified model of the head-related transfer functions, generalized to any multi-loudspeaker configuration using a transaural technique for the best pair of loudspeaker for each sound source. Although this quite simple technique does not compete with the 3D accuracy of Ambisonics or holophony (Wave Field Synthesis), it is very flexible (no specific loudspeaker configuration) and suitable for a large audience (no hot-spot effect) with sufficient sound quality.

The resulting software system is able to separate 5-source stereo mixtures (read from audio CD or 16-bit PCM files) in real time and it enables the user to remix the piece of music during restitution with basic functions such as volume and spatialization control. The system has been demonstrated in several countries with excellent feedback from the users / listeners, with a clear potential in terms of musical creativity, pedagogy, and entertainment.

For now, the mixing model imposed by the informed source separation is generally over-simplistic when professional / commercial music production is at stake. Extending the source separation technique to high-quality convolutive mixing is part of our future research.

As shown in [2], the model we use for the spatialization is more general, and can be used as well to localize audio sources. Thus we would like to add the automatic detection of the speaker configuration to our system, from a pair of microphones placed in the audience, as well as the automatic fine tuning of the spatialization coefficients to improve the 3D sound effect.

Regarding performance, lots of operations are on separated data and thus could easily be parallelized on modern hardware architectures. Last but not least, we are also porting the whole application to mobile touch devices, such as smart phones and tablets. Indeed, we believe that these devices are perfect targets for a system in between music listening and gaming, and gestural interfaces with direct interaction to move the sources are very intuitive.

Acknowledgments.

This research was partly supported by the French ANR *(Agence Nationale de la Recherche)* DReaM project (ANR-09-CORD-006).

References

1. M. Parvaix and L. Girin, "Informed source separation of underdetermined instantaneous stereo mixtures using source index embedding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, 2010.
2. J. Mouba, S. Marchand, B. Mansencal, and J.-M. Rivet, "RetroSpat: a perception-based system for semi-automatic diffusion of acousmatic music," in *Proceedings of the Sound and Music Computing (SMC) Conference*, Berlin, 2008, pp. 33–40.
3. M. Parvaix and L. Girin, "Informed source separation of linear instantaneous under-determined audio mixtures by source index embedding," *IEEE Transactions on Audio, Speech, and Language Processing*, accepted, pending publication 2011.
4. O. Yılmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
5. P. Bofill and M. Zibulevski, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.
6. P. O'Grady, B. A. Pearlmutter, and S. Rickard, "Survey of sparse and non-sparse methods in source separation," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 18–33, 2005.
7. M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.
8. J. Princen and A. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 64, no. 5, pp. 1153–1161, 1986.
9. S. Araki, H. Sawada, and S. Makino, *K-means based underdetermined blind speech separation.* S. Makino et al. (Eds), Blind Source Separation, Springer, 2007, pp. 243–270.
10. S. Araki, H. Sawada, R. Mukai, and S. Makino, "Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors," *Signal Processing*, vol. 87, no. 8, pp. 1833–1847, 2007.
11. E. Vincent, R. Gribonval, and M. D. Plumbley, "Oracle estimators for the benchmarking of source separation algorithms," *Signal Processing*, vol. 87, no. 2007, pp. 1933–1950, 2007.
12. B. Chen and G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.
13. J. Pinel, L. Girin, C. Baras, and M. Parvaix, "A high-capacity watermarking technique for audio signals based on MDCT-domain quantization," in *International Congress on Acoustics (ICA)*, Sydney, Australia, 2010.
14. J. Pinel, L. Girin, and C. Baras, "A high-rate data hiding technique for uncompressed audio signals," *IEEE Transactions on Audio, Speech, and Language Processing*, submitted.

15. ISO/IEC JTC1/SC29/WG11 MPEG, "Information technology—Generic coding of moving pictures and associated audio information—Part 7: Advanced Audio Coding (AAC)," IS13818-7(E), 2004.
16. T. Thiede, W. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. Beerends, and C. Colomes, "PEAQ - the ITU standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, vol. 48, no. 1, pp. 3–29, 2000.
17. ITU-R, "Method for objective measurements of perceived audio quality (PEAQ)," Recommendation BS.1387-1, 2001.
18. V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, 2001, pp. 99–102.
19. J. W. Strutt (Lord Rayleigh), "On the acoustic shadow of a sphere," *Philosophical Transactions of the Royal Society of London*, vol. 203A, pp. 87–97, 1904.
20. ——, "Acoustical observations i," *Philosophical Magazine*, vol. 3, pp. 456–457, 1877.
21. J. Blauert, *Spatial Hearing*, revised ed. Cambridge, Massachusetts: MIT Press, 1997, translation by J. S. Allen.
22. H. Viste, "Binaural localization and separation techniques," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Switzerland, 2004.
23. R. S. Woodworth, *Experimental Psychology*. New York: Holt, 1954.
24. G. F. Kuhn, "Model for the interaural time differences in the azimuthal plane," *Journal of the Acoustical Society of America*, vol. 62, no. 1, pp. 157–167, 1977.
25. R. E. Berg and D. G. Stork, *The Physics of Sound*, 2nd ed. Prentice Hall, 1994.
26. H. Bass, L. Sutherland, A. Zuckerwar, D. Blackstock, and D. Hester, "Atmospheric absorption of sound: Further developments," *Journal of the Acoustical Society of America*, vol. 97, no. 1, pp. 680–683, 1995.
27. *ISO 9613-1:1993: Acoustics – Attenuation of Sound During Propagation Outdoors – Part 1: Calculation of the Absorption of Sound by the Atmosphere*, International Organization for Standardization, Geneva, Switzerland, 1993.
28. C. Tournery and C. Faller, "Improved time delay analysis/synthesis for parametric stereo audio coding," *Journal of the Audio Engineering Society*, vol. 29, no. 5, pp. 490–498, 2006.
29. J. M. Chowning, "The simulation of moving sound sources," *Journal of the Acoustical Society of America*, vol. 19, no. 1, pp. 2–6, 1971.