Audio Engineering Society
Convention Paper

Presented at the 118th Convention
2005 May 28–31 Barcelona, Spain

Descriptor-based spatialization

Jérôme Monceaux [1], François Pachet [2], Frédéric Amadu [1], Pierre Roy [2], and Aymeric Zils [2]

[1] Arkamys, Paris, France, 75008
jmonceaux@arkamys.com, famadu@arkamys.com

[2] Sony CSL Paris, France, 75005
pachet@csl.sony.fr, roy@csl.sony.fr, zils@csl.sony.fr

ABSTRACT

The translation of monophonic soundtracks to new audio formats is the object of a growing demand particularly from the DVD producers. However operations like "upmixing" a monophonic track to a multi-channel format are time-consuming tasks for the sound engineer, who has to choose, adapt and tune different spatialization tools. In order to simplify the upmix, we introduce a new spatialization approach based on the use of perceptive descriptors. It consists in automatically detecting audio source characteristics, based on perceptive aspects and exploiting these characteristics to control spatialization tools. This paper presents an experimentation driven by Arkamys and Sony CSL Paris to evaluate the interest of this technique, to address the needs of sound engineers in the field of spatialization. EDS [1,2], a generic audio information extractor, was used to control the Arkamys spatializer [3]. We describe the aim and interest of this descriptor-based approach and discuss its performances and limitations.

1. THE ARKAMIX SPATIALIZATION TOOL

1.1. Presentation

Arkamix is a multipurpose spatialization toolbox developed by Arkamys. Its application for old movie spatialization must be realized in respect of the original soundtrack meanings. This transformation must be downmix compatible in mono and stereo. The Arkamix toolbox was made to simplify digital audio effects design and tuning. There are three different layers in Arkamix:

- C++, audio effects and tools,
- Graphic User Interface (GUI),
- XML Configuration file.

The GUI and the audio effects are completely driven by the XML configuration files. The GUI is able to run without the audio process and reciprocally. XML files are split in two categories: block based description of the process, and controlled values. At the startup, both GUI and process read XML files to execute a self configuration. Thanks to this architecture, Arkamix is

flexible software for audio effects designers and sound engineers. In our case, extractors from EDS (Extractor Discovery System) interact with the audio process, in complement of the original Arkamix GUI.

We have developed a specific Arkamix spatialization toolbox taking into account sound engineers techniques. We have designed generic tools such as: equalization, multi-channel compressor, limiter, panoramic mixer, etc. In complement, we added more specific tools like: mono extractor, surround extractor, short stereo and multi-channel reverberation.

Usually, we control these algorithms by a graphic user interface, but in this case, we substitute it with a control module, which is designed to translate sources identification generated by the extractor from EDS to a series of control commands. The driven controls are exactly the same as the standard controls driven by sound engineer.

Control values are principally Boolean values that answer a simple question "Is Source A detected" for instance. This information can be translated to Boolean value such as "bypass effect #1", a linear value "Control Value B9 go to value 1.0 smoothly", or it can be a mix of them. It can also be a complex command such as: “bypass effect #1, Control Value B9 go to value 1.0 smoothly AND surround level go to 0 quickly”.

1.2. Reverberation

The reverberation is based on the Arkamys Sound Process[3]. Several sets of IR and BRIR (Binaural Room Impulse Responses) were recorded in different places (studio, church, stairs, etc.) by MLS or sweep method, in order to create a huge database of virtual acoustic environments.

These IR have been derived, after normalization and time alignment, to long FIR filters. Then the output sound is computed using a fast convolution algorithm. Thanks to using standard reverberation controls, the characteristics of the any original reverberation (EQ, color, length, pre-delay, etc.) can be modified in real time in order to create a new one.

1.3. UpMix 5.1

UpMix includes a centre extraction and a surround extraction functions in order to transform a stereo signal into a 5.1 multi-channel signal as follows:

- centre extraction (C),
- surround extraction (Ls Rs),
- subtraction of C from the original stereo to create L and R
- subwoofer signal is the sub bass part of the centre (LFE).

1.3.1. Monophonic component extraction

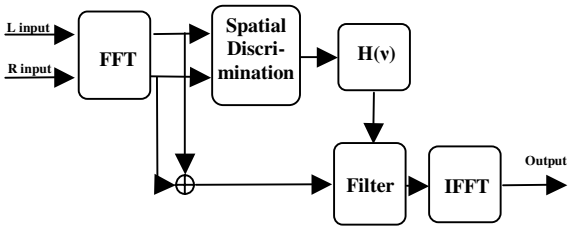

Figure 1 Centre extraction

The “spatial discrimination” is performed in order to compute a transfer function: H(ν). This transfer function is used to extract the centre component from the stereo signal downmixed in mono.

Time to frequency transformation (FFT) and frequency to time transformation (IFFT) are based on the overlap and add method.

Our spatial discriminator is based on a spectral comparison analysis between left and right channels.

If a “monophonic frequency” is present in the stereophonic signal, the spectrum difference between left and right will cause a minimum for this frequency. By analyzing these minima, we can compute H(ν) in order to extract the monophonic frequency (1). By tuning a *threshold* discrimination level, we can tune an extraction tolerance.

$$H(\nu) = \frac{\left(treshold - \left|\mathrm{fft}(L(n),\nu) - \mathrm{fft}(R(n),\nu)\right|\right)}{threshold} \qquad (1)$$

And if $H(\nu) \leq 0$ then $H(\nu) = 0$

In the frequency domain, the filter is a simple complex multiplication. After filtering, a correction bloc used to avoid out of phase artifacts performs the following operations (2):

$$D(\nu) = \min\left(\left|\text{fft}(L(n),\nu)\right|, \left|\text{fft}(R(n),\nu)\right|\right)$$

if $$\left|\text{fft}(C(n),\nu)\right| - D(\nu) \geq 0$$

then $$\text{fft}(C(n),\nu) = \frac{\text{fft}(C(n),\nu) \cdot D(\nu)}{\left|\text{fft}(C(n),\nu)\right|} \quad (2)$$

1.3.2.Surround extraction

As shown above, the subtraction of the left and right channels cancels the mono "in phase" components. By performing the addition of the stereo channels, we cancel the mono "out of phase" components. Thus, we are able to extract "out of phase" components from the original signal using the same method as the centre extraction for the surround filter Hs(v) computation.

In multi-channel 5.1 mix it is known that surround channels can include the "out of phase" part of the sound and centre are basically the mono component.

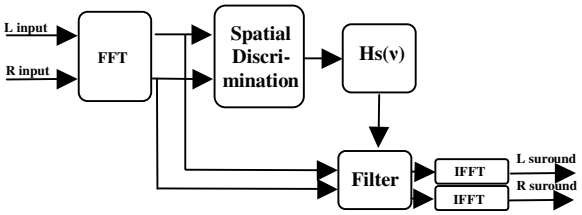

Figure 2 Surround extraction

2. EXTRACTING DESCRIPTORS

EDS (Extractor Discovery System) [1,2] is a generic scheme for extracting arbitrary high-level audio descriptors from audio signals. It is able to automatically produce a fully-fledged audio extractor (an executable program) from a database of labeled audio examples. It uses a supervised learning approach. Its main characteristics are that it finds automatically optimal audio features adapted to the problem at work, using a combination of genetic programming and combinatorial search techniques. Then resulting features are fed to a learning model such as a Gaussian Mixture Model (GMM) or Support Vector Machine (SVM) to produce a fully-fledged extractor program. More precisely, the approach implemented by EDS is the following:

Descriptors are traditionally designed by combining Low-Level Descriptors (LLDs) using machine-learning algorithms [4], [5], [6]. The key idea of our approach is to substitute the basic LLDs with arbitrary complex compositions of signal processing operators: EDS automatically composed operators to build features as signal processing functions that are *optimal for a given descriptor extraction task*.

The search for specific features is based on genetic programming, a well-known technique for exploring search spaces of function compositions [7]. The genetic programming engine automatically composed signal processing operators to build complex functions arbitrarily.

Each built feature is given a fitness value which represents how well the function performs to extract a given descriptor on a given training database. This fitness is computed as the Fisher or Pearson criteria, depending on the nature of the problem (classification or regression).

The evaluation of a feature is very costly, as it involves complex signal processing on whole audio databases. Therefore, to limit the search, a set of heuristics are introduced to improve the a priori relevance of the created features, as well as rewriting rules to simplify features before their evaluation.

Once relevant features are computed by EDS, they are fed to a generic machine learning algorithm, either for regression or classification. Each of these algorithms carries with it a certain number of parameters such as the number of neighbors in the k-NN method, or the number of layers for the Neural Networks.

The processes of selecting the right classifier and finding the right parameters for this classifier are entirely automated in EDS.

The optimization of the model consists in a complete search on all the available classifiers for all the available parameters values. The system evaluates the

performance of the classifiers by cross-validation on the training database, using various evaluation criteria such as the rate of good classifications, the correlation coefficient, or the kappa statistic [8].

The final descriptor model is the best model found, defined by (1) a set of features, (2) a model, (3) parameters for this model.

The quality of this model is evaluated on a test database for assessing it definitively.

As in the manual approach, EDS uses a training and a testing database with labeled examples. These labels are either numeric values, such as an evaluation of the "musical energy" (between 0 and 1), or a class label, such as the "presence of voice" or not. Concerning the evaluation schemes, we apply the same rules than in the traditional approach: (1) the fitness of each feature is computed using the training database, (2) the score of all the models is computed using cross-validation on the training database, and the performance of the optimal model is evaluated with the test database.

3. DESCRIPTOR-BASED SPATIALIZATION

Automatic spatialization using descriptors is meant to be an extension of existing tools, it does not aim at replacing sound engineers. Creating a multi-channel sound from a mono or stereo track generally involves many artistic choices. However, we can infer general rules in accordance to the usual spatialization approach used by sound categories, such as the following:

- Actors voices are often located in the centre channel,
- Azimuthal sources displacement observe panoramic laws,
- Surround keeps front reverberation but not direct field,
- Each reverberation is decorrelated
- Movie music is multi-channel.

Of course, these are not hard-and-fast rules and sound engineers adapt them to the scene and to artistic choices. Our work aims at automating these rules.

Experimentation was conducted with a stereophonic French movie: « La Balance », by Bob Swaim. We choose three distinct situations:

- Upmixing mono sound track to stereo,
- Upmixing stereo track to multi-channel,
- And finally, upmixing mono sound track to multi-channel.

Sound categories were chosen as follow:

- Voices,
- Music,
- Other.

3.1. Descriptors

The first task was to segment the sound track to obtain short uniform samples of approximately 10 seconds, some of them being pure voice samples, others being a mix between voice and music, voice and background sound, and some of them being pure music or noise, without voice. Then those samples were given a label depending on whether they were containing voice, or music, or both.

Segments were then used to build a training database and a testing database for EDS. Typically, the databases (training and testing) contain 150 samples each.

EDS computes relevant features for the labeled samples using a genetic algorithm. Eventually, the best features are selected using a feature selection algorithm, in our case: forward selection.

We feed these features to different classifiers, and identify the best one which is the final extractor. Eventually, we compute the performance of the resulting descriptor on the testing database.

Finally, we link the extractor to Arkamix, adapt spatialization rules and apply the sound track process.

3.2. Spatialization algorithms

We present three experimentations that conjointly use extractors from EDS and the Arkamix toolbox on the soundtrack of « La Balance ».

First, we focused on a mono to stereo algorithm. In this case, we applied a specific equalization designed by a sound engineer on detected voices.

Then, this experimentation showed that we could upmix a stereo to a mutli-channel format with a very big attenuation of detected voices. This can be used to recreate a multi-channel soundtrack without any voice.

Finally, a mono to multi-channel algorithm is introduced. By controlling reverberation and equalization with EDS extractors, we build a 5.1 that keeps voices in the centre channel and dispatch music and unvoiced sound on every channel.

3.2.1.Mono to stereo

We transform a mono stream to a stereo stream by using a short colorless reverberation and a graphical EQ. Different presets are tuned for music, voices, or any others sources detected by extractors. Afterwards, according to the detection result, these presets are automatically loaded in the Arkamix tool.

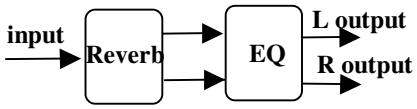

Figure 3 Mono to stereo process

We supposed that voice must be central, and that music can be stereo enhanced.

Therefore, EQ is enabled when the mix contains voice, and a specific preset is loaded with no reverberation. Otherwise, EQ is disabled and reverberation is enabled with a specific set of control values according to the detected sources (music or "other sound").

3.2.2.Stereo to multi-channel 5.1

According to the spatialization rules explained above (§3), we try to attenuate detected voices by muting only the C channel. Here, only the voice extractor is used. An attack and a release time are applied to smooth modifications of the output signal. So unvoiced multi-channel soundtrack can be extracted from the movie.

If there is some stereo reverberation in the original mix, the "UpMix tool" dispatches it on every channel except in the centre, so voice's reverberation can be heard. To cancel it, we could add a dynamic level control on L, R, Ls and Rs to attenuate the voice reverberation perception.

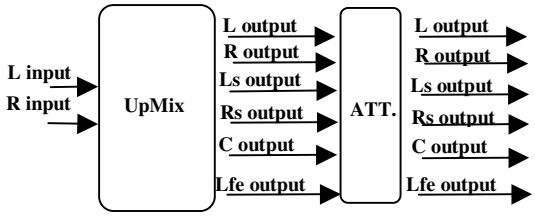

Figure 4 Stereo to multi-channel 5.1

3.2.3.Mono to multi-channel 5.1

We transform a mono stream to a multi-channel 5.1 stream by using a short colorless reverberation, a graphical EQ and the UpMix tool. Different presets are tuned for music, voices, or any other source. Afterwards, according to the detection result, these presets are automatically loaded in the Arkamix tools.

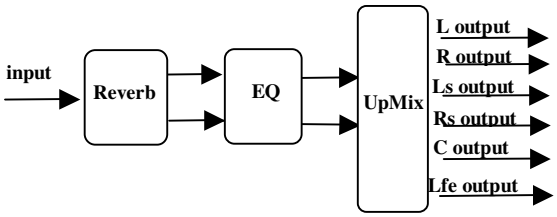

Figure 5 Mono to multi-channel 5.1

As in the "mono to stereo" process (§3.1.1), reverberation is disable when voice is detected, so that voice is kept at the centre. In any other case, reverb preset is loaded according to the detected source.

4. SUPERVISED LEARNING

We have experimented our approach on the soundtrack of « La Balance », a French movie by Bob Swaim. We have used the EDS system to generate descriptors for the following categories: voice, music and background

noise (default case) (see Section 3). Descriptors are obtained by combining Low-Level Descriptors (LLDs) using machine-learning algorithms. We will now illustrate the descriptor generation scheme on the problem of detecting the presence of human voice.

Firstly, we segment the soundtrack in order to obtain short uniform samples, of approximately 10 seconds, some of them being pure voice samples, others being a mix between voice and music, voice and background sound, and some of them being pure music or noise, without voice. Those samples are given a label depending on whether they contain voice, music, or both. We then take some samples as a training database for the EDS system, and other samples as a testing database.

Let us illustrate this approach with two specific problems: (1) voice presence and (2) music presence. In both cases, we create a training and a testing database, each database contains 150 samples. For the first problem (i.e. voice presence), 50% of the samples contain voice, and the 50% remaining don't. For the second problem (i.e. music presence), 30% of the samples contain music, and 70% don't.

We compute acoustic features on the training database, by combining LLDs using a genetic-programming algorithm. Here are some of the features found by the EDS system for the voice presence detection:

Log10 (Iqr (SpectralKurtosis (Hamming (SplitOverlap (x, 2048.0, 0.5)))))

Log10 (Mean (Square (Derivation (Sqrt (SpectralSkewness (Hamming (Split (x, 3517.0)))))))))

Iqr (MaxPos (SplitOverlap (x, 2048.0, 0.5)))

Max (SpectralCentroid (LpFilter (FilterBank (x, 4.0), 16060.0)))

Where "x" denotes the input signal, i.e. the audio sample processed.

EDS computes many acoustic features and assesses their fitness, using the Fisher separability criteria evaluated on the training database. Then, EDS selects the best features using a feature selection algorithm (in our case forward selection).

EDS finally feed these features to different classifiers, and identify the best one. The list of classifiers used is provided by the Weka tool kit [9] and includes Naïve Bayesian classification, Support Vector Machines, Neural Nets, Kernel Density Classification, KNN, etc. The score of a classifier is obtained by 10-fold cross-validation on the training database: the model is trained on 90% of the database, and evaluated on the remaining 10%. This operation is performed 10 times, and the average performance is computed.We compute the performance of the resulting descriptor on the testing database, which has the same characteristics than the training one.

In our case, the best score was obtained by retaining the seven best features, whose list follow, and a KNN classifier.

Iqr (MaxPos (SplitOverlap (x, 2048.0, 0.5)))

Max (SpectralCentroid (LpFilter (FilterBank (x, 4.0), 16060.0)))

Min (Power (SpectralKurtosis (SplitOverlap (x, 2048.0, 0.5)), 4.0))

Log10 (Iqr (SpectralKurtosis (Hamming (SplitOverlap (x, 2048.0, 0.5)))))

Variance (SpectralSkewness (SplitOverlap (x, 2048.0, 0.5)))

Power (Log10 (Iqr (MaxPos (Square (Bartlett (Split (x, 9135.0)))))), -1.0)

Log10 (Mean (Square (Derivation (Sqrt (SpectralSkewness (Hamming (Split (x, 3517.0)))))))))

The KNN classifier created a descriptor which classifies 96% of the samples of the testing database correctly.

For the music detection problem, we came up with the following features:

Power (Abs (Log10 (Mean (Rms (Blackman (Split (Autocorrelation (x), 860.0)))))), 3.0)

Sqrt (Min (Holes (Holes (Sum (Blackman (Split (Autocorrelation (x), 860.0)))))))

Rms (Range (Blackman (Split (Autocorrelation (x), 860.0))))

Power (Log10 (Abs (Percentile (SpectralSpread (Blackman (Split (Autocorrelation (x), 860.0))), 83.0))), 3.0)

The best classification model EDS found for this problem is SVM, which yields a 95% success ratio on the testing database.

5. PERFORMANCES AND LIMITATIONS

The best extractors found by EDS for the voice and music detection problems have a success rate slightly above 95% on the testing databases, which is quite satisfactory.

Indeed, in a production context, 5% of errors is acceptable, for instance concerning the spatial enhancement of music. In other cases though, 95% is still not good enough. For instance, the processing of voice must be near perfect, to ensure intelligibility of the spoken content.

We expect these results to be substantially improved if we use larger, better designed training databases. Moreover, we plan to improve the algorithm used to build relevant features, in particular by adding audio signal operators that are specific to voice detection such as “formant”.

Although some preliminary operations still have to be performed manually (e.g. segmenting the soundtrack, indexing the training databases), the approach we presented, based on the combined use of EDS and Arkamix, is a concrete step towards automating spatialization. The next step will be to integrate an automatic segmentation module in our system and to have built-in extractors for most of the sound categories classically involved in movies, i.e. voice, music, background noise, etc. For less standard sounds, such as specific character voices or specific sounds, extractors will still have to be designed manually.

6. CONCLUSION

We have presented an innovative approach to the problem of remastering movie soundtracks in an automatic manner.

This problem has become important with the commercial success of home cinema devices (5.1 multi-channel systems). Adapting the soundtrack of an old movie to those new audio systems requires sound engineers to spend a huge amount of time watching the movie, segmenting it and mixing every sequence manually.

This task can be partly automated with our process. It is based on two software systems: the Arkamix toolbox, which features audio effects and spatial discrimination of sound, and EDS which performs automatic sound descriptor discovery.

The combination of these two systems results in an approach that is automatic, if not fully, for most of the work involved in the remastering process. Besides, it allows achieving operations that were not possible so far, e.g. automatic channel separation, extraction of a voice channel from a single audio source, which could be used to translate movie soundtracks.

The approach presented here could be used to automatically remaster old soundtracks to adapt them to more modern formats (e.g. mono to stereo, or multi-channel). This would free sound engineers from tedious operations of remastering, allowing them to focus on more creative and artistic issues, for which decisions have to be made, (e.g. soundtracking).

7. ACKNOWLEDGEMENTS

Many thanks to Jean-Michel Raczinski, Julien Ricard, for all their help and support during this project.

8. REFERENCES

[1] Zils, A. and Pachet, F. Automatic Extraction of Music Descriptors from Acoustic Signals using EDS. Proceedings of the 116th AES Convention, Berlin, May 2004.

[2] Zils, Aymeric, Pachet, François (2003). Extracting automatically the perceived intensity of music titles. Proceedings of 6th International Conference on Digital Audio Effects (DAFX03), London, UK, September 8-11.

[3] Raczinski, J.-M. and Monceaux, J. and Vielledent, G.C. Sound Field Management with the Arkamys

Sound Process. Proceedings of the 114th AES Convention, Amsterdam, March 2003.

[4] Scheirer, Eric D. and Slaney, Malcolm (1997). Construction and evaluation of a robust multifeature speech/music discriminator. Proc. ICASSP '97.

[5] Herrera, P. Yeterian, A. Gouyon, F. (2002) Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. Proceedings of 2nd International Conference on Music and Artificial Intelligence, Edinburgh, Scotland.

[6] Peeters, Geoffroy and Rodet, Xavier (2002). Automatically selecting signal descriptors for sound classification. Proc. of ICMC, Goteborg (Sweden).

[7] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press, 1992.

[8] K. Fukunaga (1990). Statistical Pattern, Recognition. Academic Press.

[9] Witten, I.H., Eibe, F. and Kaufmann, M. (1999) Data mining: practical machine-learning tools with Java implementations. Morgan Kaufmann, October.