Zsa.Descriptors: a library for real-time descriptors analysis

Mikhail Malt [*], Emmanuel Jourdan[†]
[*] IRCAM, Paris, France, Mikhail.Malt@ircam.fr
[†] IRCAM, Paris, France, Emmanuel.Jourdan@ircam.fr

I. Introduction

In the past few years, several strategies to characterize sound signals have been suggested. The main objective of these strategies was to describe the sound [1]. However, it was only with the creation of a new standard format for indexing and transferring audio MPEG 7 data that the desire to define audio data semantic content descriptors came about [2, p.52]. The widely known document written by Geoffroy Peeters [1] is an example where, even if the goal announced is not to carry out a systematic taxonomy on all the functions intended to describe sound, it does in fact systematize the presentation of various descriptors.

A. Descriptors Today

A large percentage of the uses for descriptors concern primarily indexing and browsing contents of sound databases or for re-synthesis of sounds, such as in telephone transmissions.

Our interest concerns the use of the analysis of descriptors in real-time for the creation and analysis of contemporary music. In this domain, with the exception of the fundamental frequency and the energy of the sound signal, the use of spectral descriptors is still rare. It is important nonetheless, to examine the experiments on computer-assisted improvisation carried out by Assayag, Bloch, and Chemillier [3] [4] and the developments in "concatenative synthesis" by Diemo Schwarz [5] where the analysis of a variety of sound descriptors is used to control re-synthesis.

B. Descriptors and Music Composition

The fact that descriptors are rarely used in contemporary music compositions is due to several factors including:

- The lack of knowledge of the relationships between descriptors and the pertinent perceptual characteristics of the sound for use in musical composition;
- The fact that one descriptor is not sufficient in order to characterize a complex "sound state" such as that of a note played "live." Recent studies [6] show how the composed functions of descriptors are more effective in recognizing the characteristics of a given sound signal than the use of one descriptor at a time;
- The lack of a large choice of descriptors in real-time so that artists can test them and learn to use them.

II. Real-Time Environments and Descriptors

Among the most widely used software environments for real-time musical performances are *SuperCollider* [7], *PureData* [8], and *Max/MSP* [9]. Max/MSP offers the largest selection of tools to work with sound descriptors. Currently, several libraries offering analyses of descriptors are available in Max/MSP. The best known of these environments include the library by Tristan Jehan [10] [11] (pitch~, loudness~, brightness~ , noisiness~, bark~, analyzer~, shifter~, segment~, beat~), the iana~ object of Todor Todoroff, the yin~ object implemented by Norbert Schnell, according to the Cheveigné and Kawara model [12], the *FTM/Gabor* object library [13] [14] that enables development of descriptors, and finally the classic fiddle~ and bonk~ by Miller Puckette [15].

However, a large number of the descriptors offered are, as we have already mentioned, based on the recognition of the fundamental frequency and the energy. The only exceptions are the descriptors offered in the *Gabor* library, but they do not cover yet a large set of descriptors.

III. The First Descriptors Set Available in Zsa.Descriptors

The Zsa.Descriptors library is intended to provide a set of audio descriptors specially designed to be used in real-time. This objects collection encloses a sound descriptors set coming from the MPEG-7 Descriptors, outlined by Peeters [1], algorithms for peak search from Serra [16] and some ideas from the Computer Assisted Composition developments realized by the Musical Representation Team at Ircam. In the next paragraphs we will describe some of this tools, already developed in the Zsa.Descriptors library.

A. Spectral Centroid (brightness)

This is a very well known descriptor. The Spectral centroid is the barycentre of spectra, computed as follow:

$$\mu = \frac{\sum_{i=0}^{n-1} f[i]a[i]}{\sum_{i=0}^{n-1} a[i]}$$

Where:

n, is the half of the fft window size

i, the bin index

$a[i]$, is the amplitude of the bin i, the real part of the FFT calculus

$f[i]$, is the frequency of the bin i. where

$$f[i] = i * \frac{sample\ \ rate}{fft\ \ window\ \ size}$$

and

μ, is the spectral centroid in hertz.

B. Spectral Spread (spectral centroid variance)

As usual, we consider the spectral centroid as the first moment of spectra, considered as a frequency distribution, which is related with the weighted frequency mean value. The spectral spread is the second moment, i.e., the variance of the mean calculated above.

$$v = \frac{\sum_{i=0}^{n-1} (f[i] - \mu)^2 a[i]}{\sum_{i=0}^{n-1} a[i]}$$

C. Spectral Slope

The spectral slope is an estimation of the amount of spectral magnitude decreasing, computed by a linear regression on the magnitude spectra.

$$slope = \frac{1}{\sum_{i=0}^{n-1} a[i]} \frac{n\sum_{i=0}^{n-1} f[i]a[i] - \sum_{i=0}^{n-1} f[i] \sum_{i=0}^{n-1} a[i]}{n\sum_{i=0}^{n-1} f^2[i] - \left(\sum_{i=0}^{n-1} f[i]\right)^2}$$

D. Spectral Decrease

The spectral decrease meaning is similar to spectral slope, representing the amount of spectral magnitude decreasing. According to Peteers [1], this formulation comes from perceptual studies and it is supposed to be more correlated to human perception.

$$decrease = \frac{\sum_{i=0}^{n-1} a[i] - a[1]}{\sum_{i=2:K}^{n-1} a[i](i-1)}$$

E. Spectral Roll-Off

The spectral roll-off point is the frequency $f_c[i]$ so that x% of the signal falls below this frequency. "x" is took as 0.95 as default value. The roll-off point is calculated as follow:

$$\sum_{i=0}^{f_c[i]} a^2[f[i]] = x\sum_{i=0}^{n-1} a^2[f[i]]$$

where:

$f_c[i]$, is the roll-off point and

x, the roll_off energy percent accumulated.

F. Sinusoidal model based on peaks detection

We have based the calculus of our algorithm on the widely known method defined by Smith&Serra [16] [18, p. 38-48], where a peak is defined as a local maximum in the real magnitude spectrum $a_k[i]$. "k" is the frame index. As not all the peaks are equally important in the spectrum, we have used a sliding five points window to scan the magnitude spectra, avoiding undesired peaks. For each 5 magnitudes vector we check for the third point $a_k[2]$, and for a given threshold value ε_t, we compute:

$a_k[2] = \max\{a_k[0], a_k[1], ..., a_k[4]\} \wedge a_k[2] > \varepsilon_t$.

If the condition is true, then $a_k[2]$ becomes a peak.

A parabolic interpolation is then applied on the three adjacent points, $a_k[1], a_k[2], a_k[3]$.

Solving the parabola peak location [18, p. 47], a coefficient "p" of the "j" peak is then calculated:

$$p_j = \frac{1}{2} \frac{a_k[1] - a_k[3]}{a_k[1] - 2a_k[2] + a_k[3]}$$

The true peak location (in bins) is given by:

$$i_{peak[j]} \equiv i_{a_k[2]} + p_j$$

To estimate the true magnitude we use $\boldsymbol{p}$ as follow:

$$a_{k\,peak[j]} \equiv a_k[2] - \frac{1}{4}(a_k[3] - a_k[1])p_j$$

At the end of the process we have collected a set of partials $pk_j = (f_j, a_j)$.

G. A Tempered Virtual fundamental

This descriptor was based on the harmonic histogram technic described by Jean Laroche [20, p.52-53]. We adapted this method in order to approximate the result and the research phase for the "best candidate", by a tempered musical scale with a given division.

Given a set of peaks $pk_j = \left(f_j, a_j\right)$, calculated as showed previously,

1) For each $pk_j = \left(f_j, a_j\right)$ we calculate a set of

$$pk_{jn} = \left(\frac{f_j}{n}, a_j\right), n \in N, n \subset [1,..,6].$$

2) All the $\frac{f_j}{n}$ were converted in indexes, i_{jn}, in a pitch-class space. At this level $i_{jn} \in R$.

3) The i_{jn} were approximated by a grid of discrete values multiples of q, $q \in R, q \subset [0,1]$, returning a new set of values i_{jn}^q, multiples of q. Notice that q can be seen as a half-tone division. $q = 1$, means an approximation by a half-tone, $q = 0.5$ an approximation by a quarter-tone, and so on.

4) This leads us to new couples $pk_{jn}^q = \left(i_{jn}^q, a_j\right)$.

5) Collecting all couples according with the identical i_{jn}^q, we build new couples $pk_{jn}^q = \left(i_{jn}^q, \sum a_j\right)$, where $\sum a_j$ is the sum of all a_j for the identical i_{jn}^q.

6) The best candidate to be our virtual fundamental will be the $pk_{jn}^q = \left(i_{jn}^q, \sum a_j\right)$, that maximises $\sum a_j$.

7) In the last phase, i_{jn}^q is converted, in floating point MIDI pitches or in a frequency space.

IV. The Solution Offered by Zsa.Descriptors

As was exposed previously, the main goal of Zsa.Descriptors, a library of sound descriptors and spectral analysis tools, is to expand the capabilities of sound description using the systematic approach of the MPEG7 standard, and to offer a set of truly integrated objects for the *Max/MSP* [9] graphical programming environment. In addition to sound descriptors and original analysis features, the external objects of the Zsa.Decriptors library are designed to compute multiple descriptors in real-time with both efficiency in terms of CPU usage and guaranteed synchronization. In consequence a modular approach was chosen. In the *MAX/MSP* environment context, this was made possible by sharing the expensive process of the windowed FFT.

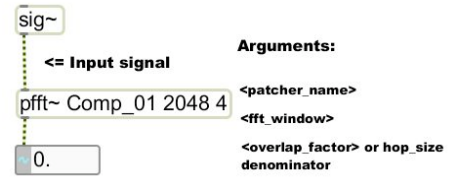

Fig. 1 the pfft~ object

MAX/MSP, actually has an object that calculates, in an efficient form, a windowed FFT, the "pfft~" facility.

As is said, in the documentation, "pfft~", is a "Spectral processing manager for patchers", i.e., an object that can load special designed "patchers". The "pfft~" object takes at least three arguments: the patcher name, the FFT window size and an overlap factor, to calculate the hop size (Fig. 1). The loaded "patcher" must also follow a general structure. This patcher must have at least an "fftin~" object. The pfft~ object manages the windowing and overlap of the incoming signal, fftin~ applies the window function (envelope) and performs the FFT. The fftin~ object takes two arguments, the "inlet assignment" and the name of the window envelop function (hanning, hamming, square and blackman are included), or the name of buffer~. It is therefore possible to use any kind of window depending on the type of sound that we want to analyse

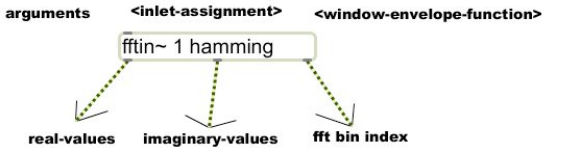

Fig. 2 the fftin~ object

Therefore, most of the objects of the library, was designed to run inside the standard *MAX/MSP* pfft~ object (Fig. 3). This strategy offers multiple advantages: modularity, efficiency, and also the ability of using the analysis directly as parameter for sound processing in the spectral domain.

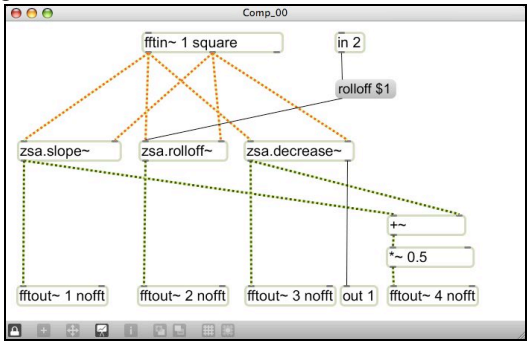

Fig. 3 Interior of the pfft~ object

Furthermore, the fact that the objects of this library can work within the Max/MSP environment either together or by themselves and the fact that they work smoothly in conjunction with other standard Max/MSP objects, makes it possible to exploit all the synchronization resources available in this environment.

V. Conclusions and Perspectives

We have presented in this paper a set of sound descriptors and signal analysis tools intended to be used in real-time as a toolbox for composers, musicologist and researchers. This will allow the use and the research on the use of sound descriptors in the fields of systematic musicology and as a tool for taking decisions in the real-time performance context. As part of the future work, we have also planed a research on the musical segmentation based on sound descriptors as a strategy to musical analysis.

The main advantage, of this library, more than the small improvements we did in some algorithms, was the modular development technique and implementation we

have used, trying to optimise the calculus by a strong integration in the MAX/MSP environment.

Of course, the work presented here still preliminary, but it will be improved with the implementation of the following list of features, which are already implemented or currently being developed: temporal variation of the spectrum, bark, inharmonicity, harmonic spectral deviation, odd to Even harmonic energy ratio, tristimulus, frame energy, harmonic part energy (this harmonic descriptors will use a monophonic F0 algorithm developed by Chunghsin YEH in his Ph.D. thesis [21]), noise part energy, and others descriptors coming from the signal processing and computer assisted composition worlds.

ACKNOWLEDGMENT

We would like to thank Richard Dudas for his fruitful remarks, comments and suggestions, Arshia Cont for his friendly support and Cyril Beros for funding the travel support.

REFERENCES

[1] G. Peeters, A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Cuidado projet report, Institut de Recherche et de Coordination Acoustique Musique (IRCAM), 2004.

[2] F. Gouyon, Extraction automatique de descripteurs rythmiques dans des extraits de musiques populaires polyphoniques, mémoire de DEA ATIAM, Université de la Méditérranée, Université Paris VI, IRCAM, Télécom-Paris, université du Maine, Ecole Normale Supérieure, ACROE-IMAG, Juillet 2000.

[3] G. Assayag, G. Bloch, M. Chemillier, A. Cont, S. Dubnov – « Omax Brothers : A Dynamic Topology of Agents for Improvization Learning », in Workshop on Audio and Music Computing for Multimedia, ACM Multimedia 2006, Santa Barbara, USA, October 2006.

[4] G. Assayag, G. Bloch, M. Chemillier – « OMax-Ofon », in Sound and Music Computing (SMC) 2006, Marseille, France, Mai 2006

[5] D. Schwarz, « Current research in concatenative sound synthesis », Proceedings of the International Computer Music Conference (ICMC), Barcelona, Spain, September 5-9, 2005.

[6] Zils A., Extraction de descripteurs musicaux: une approche évolutionniste, Thèse de Doctorat de l'Université Paris 6, Septembre 2004.

[7] SuperCollider, © James McCartney, http://www.audiosynth.com/

[8] PureData, © Miller Puckette, http://crca.ucsd.edu/~msp/Pd_documentation/

[9] Max/MSP, © Cycling74, www.cycling74.com

[10] T. Jehan, B. Schoner, « An Audio-Driven, Spectral Analysis-Based, Perceptual Synthesis Engine », in Audio Engineering Society, Proceedings of the 110th Convention, Amsterdam, The Netherlands, 2001.

[11] T. Jehan, Creating Music by Listening, PhD Thesis in Media Arts and Sciences, Massachusetts Institute of Technology, September 2005.

[12] A. De Cheveigné, H. Kawahara, « YIN, a fundamental frequency estimator for speech and music », J. Acoust. Soc. Am. 111, 1917-1930, 2002.

[13] N.Schnell et al. « FTM Complex Data Structures for Max/MSP », in ICMC 2005, Barcelona, Spain, 2005..

[14] N. Schnell, D. Schwarz, "Gabor, multi-representation real-time analysis/synthesis", Proc. of the 8th Int. Conference on Digital Audio Effects (DAFx'05), Madrid, Spain, September 20-22, 2005

[15] M. Puckette, T. Apel,. « Real-time audio analysis tools for Pd and MSP ». Proceedings, International Computer Music Conference. San Francisco: International Computer Music Association, 1998, pp. 109-112.

[16] J.O. Smith, X. Serra, "PARSHL: an analysis/synthesis program for non- harmonic sounds based on a sinusoidal representation". Proc. 1987 Int. Computer Music Conf. (ICMC'87), Urbana, Illinois, August 1987, pp. 290 -297.

[17] B. Doval, X. Rodet, "Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and HMMs." Proceedings of the ICASSP '93, 1993, pp. 221- 224.

[18] X. Serra, A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition. Philosophy Dissertation, Stanford University, Oct. 1989

[19] X. Rodet, "Musical Sound Signal Analysis/Synthesis: Sinusoidal+Residual and Elementary Waveform Models", in TFTS'97 (IEEE Time-Frequency and Time-Scale Workshop 97), Coventry, Grande Bretagne, august 1997.

[20] J. Laroche, Traitement des signaux audio-fréquences, TELECOM, Handout, Paris, France, February 1995.

[21] C. YEH, Multiple fundamental frequency estimation of polyphonic recordings, Ph.D. thesis, Université Paris 6, 2008.