PHASE VOCODER APPLICATIONS IN GRM TOOLS ENVIRONMENT

Emmanuel Favreau

Ina-GRM
Maison de Radio-France
116, avenue du président Kennedy
74220 Paris CEDEX 16
efavreau@ina.fr

ABSTRACT

This paper focuses on four real time audio processes based on the phase-vocoder, or more generally on analysis/synthesis algorithms by Fourier transformation. These effects are available as VST (Steinberg) or RTAS (Digidesign) plugins. Constraints dependent on an exclusive real-time use will be exposed as well as the related issues on algorithmics and user interface.

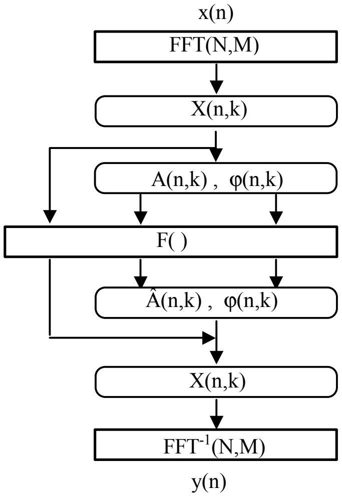
1 INTRODUCTION

The constant increase in personal computer capacities allows the development of interactive real time audio processes based on analysis/synthesis techniques such as the phase-vocoder. The possible manipulation of a frequential signal representation rather than a temporal one enables access to new categories of sound signal transformation and offers the user new interactive accesses. However these algorithms may have to handle large amounts of data which are both difficult to represent and to easily control if no simple processes, such as constant time scaling or pitch shifting, are used. After a brief reminder of the underlying principles of these algorithms, each effect will be presented in details, with a focus on the interface related issues and the way they have been solved.

2 PRINCIPLE

2.1 The phase-vocoder

The phase vocoder is based on analysis/synthetis principles and short time Fourier Transform that have been described in length in dedicated literature [1], [7], [2]. A schematic diagramm will remind the fundamental elements and the notations that will here be used.



$$X(n,k) = X_R(n,k) + jX_I(n,k) = A(n,k)\, e^{j\varphi(n,k)}$$

is the short time Fourier transform (with an N samples analysis window and M samples hop size) of the signal x(n) at the instant n, in the channel k with an amplitude A(n,k) and a phase φ(n,k). After transformation by the function F the following is obtained:

$$\hat{X}(n,k) = \hat{X}_R(n,k) + j\hat{X}_I(n,k) = \hat{A}(n,k)\, e^{j\hat{\varphi}(n,k)} = F(X(n,k))$$

then y(n) by inverse Fourier transform.
The processes will apply either on complex values X_R and X_I, either on their polar representation A et φ.

2.2 The interface constraints

These audio-processes are used as plugins within a real time environment, VST (Steinberg) or RTAS (Digidesign), taking into account some GRM Tools distinctive attributes [7], [8]. The user and host application interface constraints then are as follows :

- Exclusive real-time processing of a sequencer issued audio stream. It is thus impossible to achieve audio process such as time-stretching.
- None or limited prior process configuration. The effect must be available as soon as the plug-in is inserted in the host application.
- Only a limited number of controls can be displayed so as not to overflow the interface.

- All controls must be simultaneously displayed. The control values must be visible at once.
- All controls can be manipulated, automated and interpolated with, of course, no discontinuity nor incoherence.
- The process may be used without prior knowledge of the used algorithm. It is surely the harder constraint since the user must be conscious that he is working with a spectrum to get optimum use of the proposed effect.

It is not possible to take all these constraints at once into account and develop only one plugin that would allow exploitation of all the phase-vocoder capacities, while maintaining a simple and intuitive interface.
Four effects have been developped, each one focusing on a different aspect of the phase vocoder while using the most fitted control and visualization items. The effects are as follows :

- **Equal**, a 31 bands equalizer (amplitude modification)
- **Contrast**, differentiated level amplitude control (amplitude modification)
- **FreqShift**, a pitch shifter linked to a transposer (frequency modification)
- **FreqWarp**, an arbitrary redistribution of frequencies (frequency modification)

3 THE EFFECTS

First will be presented the effects that only modify the amplitude of various spectral components, then those modifying their respective frequency.

3.1 Equal

3.1.1 Principle

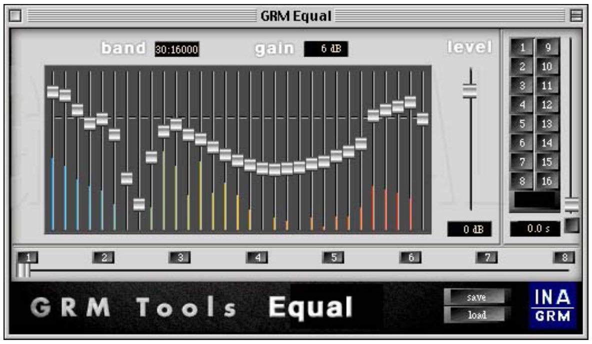
The most simple process that can be achieved is to multiply the amplitude of each frequency components with a constant value. The transformation function is then as follows :

$$\hat{X}(n,k) = g(k)X(n,k)$$

a filter is obtained, the frequency response of which is defined by the coefficients g(k). It is a circular convolution that can create a time-aliasing [1]. However, several hearing tests have showed that this simplified technique gives good results.

3.1.2 Interface

The required parameters are the size N of the analysis window, the value M of the hop size and the coefficents g(k). In order to define these coefficients, a representation by a graphic equalizer with 31 third octave bands has been chosen, in agreement with the ISO standards. The user is confronted with a regular filter representation that requires no special learning. The values g(k) are directly derived from the position of the sliders that vary from +12dB to -∞ dB. The size of the analysis window has been set on 2048 samples, a sufficient enough value to obtain the required resolution for a third octave division, and the hop size is 1024 samples. These two values cannot be modified.



3.2 Contrast

3.2.1 Principle

This effect is an adaptation of a dynamic multiband control [3]. The amplitude, in decibels, of each frequency band is modified by a arbitrary G function:

$$\log_{10}(\hat{A}(n,k)) = G(\log_{10}(A(n,k)))$$

The multiplying coefficients g(n,k) are directly induced:

$$g(n,k) = \frac{\hat{A}(n,k)}{A(n,k)} = 10^{G(\log_{10}(A(n,k))) - \log_{10}(A(n,k))}$$

It is only necessary to tabulate the function

$$g'(x) = 10^{G(x)-x}$$

and address it with the logarithm of the amplitude A(n,k) to obtain g(n,k):

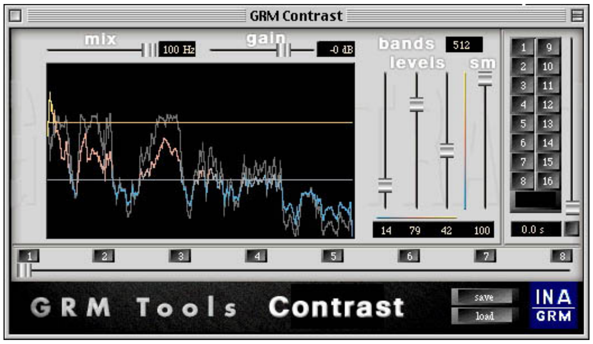
$$g(n,k) = g'(\log_{10}(A(n,k)))$$

and

$$\hat{X}(n,k) = g(n,k)X(n,k)$$

3.2.2 Interface

Required parameters are the size N of the analysis window, the value M of the hop size and the function G. The direct edition of this function has no significant value since the curve is not seemingly related to the result of the process. So as to provide the user with a more intuitive interface, a different solution was adopted. The whole of the spectral components are divided into three sub-groups that match low, mid and high amplitudes. These zones are defined by two horizontal cursors on the graphic representation of the signal spectrum. The amplitudes of each sub-group are then independantly controled via the three *levels* potentiometers. The size of the analysis window varies from 256 to 16384 samples. The hop size is fixed to N/2.



GRM Contrast

mix 100 Hz

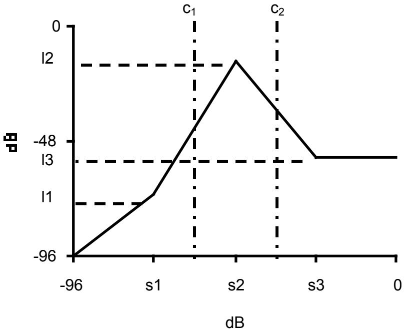
gain -0 dB

bands 512

levels 14 79 42 100

sm 0.0 s

The G function of amplitudes transformation is thus obtained:



s1, s2 et s3 are the centres of the three areas defined by the cursors c1, c2 ; l1, l2 and l3 are the three levels defined by the *levels* potentiometers.

3.3 FreqShift

3.3.1 Principle

This effect allows the production of pitch-scaling effects by a multiplication of each spectral components frequency by a scalar constant, and pitch-shifting effects by addition of a constant frequency to each spectral component. Two strategies are available to obtain the resulting spectrum:

- Browsing the source spectrum bands and achieving transformation for each one towards the resulting spectrum.
- Browsing the resulting spectrum bands and finding for each one the matching component in the source spectrum by inverse transformation.

These two solutions are equivalent regarding the frequency shift, but not as far as the scaling is concerned. In the last case, the spectrum will be compacted (lower scale transposition) or stretched (higher scale transposition). The number of significant bands will be different in the source spectrum and in the resulting spectrum. With a simple algorithm with no spectral interpolation, the first method produces a lesser amount of reconstruction artefacts ; this is the reason why it has been chosen.

The algorithm is thus:

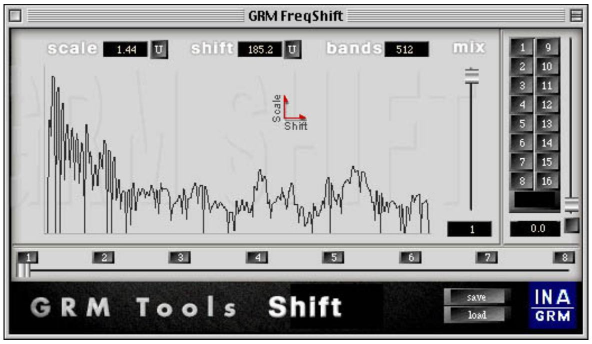
$$\hat{k} = k \cdot scale + offset$$

$$\hat{A}(n,\hat{k}) = A(n,k)$$
$$\hat{Fr}(n,\hat{k}) = Fr(n,k) \cdot scale + offset$$

$Fr(n,k)$ is the frequency in the k band obtained from φ(n-M,k) et φ(n,k) by means of a traditional phase unwrapping algorithm [1]. This very basic algorithm has several advantages : if k is negative or superior to N/2, the resulting spectrum is not modified, which automatically solves the problem of spectrum foldover that might occur with resampling based pitch-shifting or pitch scaling algorithms.

3.3.2 Interface

Required parameters are the N size of the analysis window, the M hop size value, and the scaling and offset values. A bidimensional potentiometer has been chosen for the control of scaling and offset parameters that respectively vary from 0.5 to 2 and from –4000 to +4000 Hz on a logarithm scale. The size of the FFT window varies from 256 to 8192 samples. The hop size is set on N/4 which allows a good frequency estimate during the unwrapping phase algorithm.



GRM FreqShift

scale 1.44 U

shift 185.2 U

bands 512

mix 1

0.0

3.4 FreqWarp

3.4.1 Principle

This algorithm generalizes the former one. The spectrum is not linearly modified through addition and multiplication by a constant value but arbitrarily by means of a G transfer function:

$$\hat{Fr}(n,k) = G(Fr(n,k))$$

which leads to the former process, stating :

$$scale = 1$$

and

$$offset = offset(n,k) = G(Fr(n,k)) - Fr(n,k)\ .$$

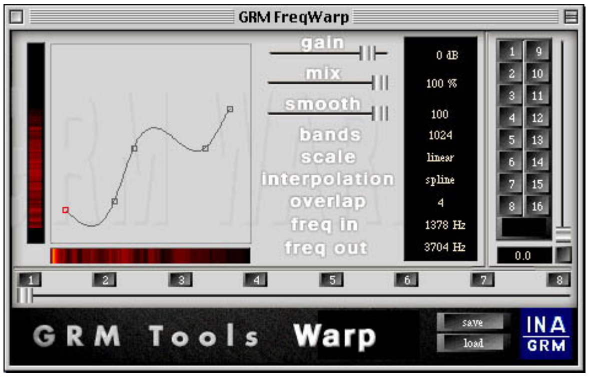
which amounts to:

$$\hat{k} = k + offset(n,k)$$

$$\hat{A}(n,\hat{k}) = A(n,k)$$
$$\hat{Fr}(n,\hat{k}) = Fr(n,k) + offset(n,k)$$

3.4.2 Interface

The required parameters are the N analysis window size, the M hop size value and the G transfer function. This function is directly edited by means of breakpoints. A source and resulting spectrum representation provides the user guidance for the definition of this function. Three frequency scales are available (one linear and two logarithmic) The logarithmic scales provide a better editing accuracy for the transfer function in the lower part of the spectrum. Breakpoints may be interpolated linearily or by splines. The size of the analysis window varies from 256 to 8192 samples. The hop size varies from N/2 to N/8. The N/2 value is not sufficient for a precise estimate of frequencies but the choice of this value was made possible to match the calculating capacity of the processor. Indeed, this treatment is rather complex and problems might occur while using it in real time on low powered equipement.



gain 0 dB
mix 100 %
smooth 100
bands 1024
scale linear
interpolation spline
overlap 4
freq in 1378 Hz
freq out 3704 Hz

4 CONCLUSION

In this article were presented four real time audio processes using the phase vocoder algorithm. Simple implementations were preferred ; however a few problems occurred. The following aspects remain to be improved :

- Stereo sources processing. The processes that modify each analysis band frequency (such as Warp and FreqShift) loose the phase relationships between the two channels, if those are independantly processed. This induces a loosening of the stereo image spatial coherence, which widens and gives the impression of stemming from two independant sources. An evident solution to the problem is to avoid direct process of the stereo signal, but process the matriced signal (sum and difference). The problem remain, but appears as a tightening of the stereo image and a transformation towards a mono signal, which is less annoying from a perceptive point of view. Other directions remain to be explored, such as joint estimate of frequencies or resynchronization of phases.
- Usual artefacts of the phase vocoder (phasing effect, frequency smearing, reverberation …) are present in these simplified implementations. Solutions were proposed [4] [5] and will need testing.

These defects or artefacts are not necessarily to be avoided. Hence, if they prove to be controlable, they indeed enrich the possible transformations made possible by the algorithm, even if from a purely mathematical point of view, they are reconstruction defaults. For instance the “ phasing artefacts ” on percussive sounds deeply modify the attacks. Just as a very short analysis window, about 128 samples for instance, produces an “ aquatic ” sound, with “ bubbling ” effects. Yet, these defects have to be kept under control and the user interface should allow their manipulation . This provides a research directions which needs to be taken into account.

5 REFERENCES

[1] D. Arfib, F. Keiler, U. Zoelzer. Time-frequency Processing, to be published in DAFX*: Digital Audio Effects*, 2001.

[2] M.R. Portnoff. Implementation of the digital phase vocoder using the fast fourier transform. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, 41(7) :2429-2438,1993

[3] U. Zolzer. *Digital Audio Signal Processing*, chap 7, John Wiley & Sons 1997

[4] J. Laroche, M. Dolson. New Phase-Vocoder Techniques for Real-Time Pitch Shifting, Chorusing, Harmonizing, and Other Exotic Audio Modifications. *J. Audio Eng. Soc*, Vol. 47, No. 11, 1999

[5] M. Puckette. Phase-locked Vocoder. *IEEE ASSP Conference on Applications of Signal Processing to Audio and Acoustics*, Mohonk 1995.

[6] S. Sprenger. Stephan M. Sprenger’s Audio DSP Pages. http://www.dspdimension.com/

[7] E. Favreau, Les Outils de Traitement GRM Tools. *Journées d’Informatique Musicale 1998* La Londe-les-Maures 1998, (pp. E4.1-E4.4).

[8] Y. Geslin, Sound and Music Transformation Environments: A twenty-year Experiment at the “Groupe de Recherches Musicales”. *Proc. Workshop on Digital Audio Effects (DAFx-98),* Barcelona, Spain, (pp. 241-247).