

Spectral analysis based synthesis and transformation of digital sound: the ATSH program

Oscar Pablo Di Liscia¹, Juan Pampin²

¹Carrera de Composición con Medios Electroacústicos, Universidad Nacional de Quilmes(Buenos Aires, Argentina)

²Center for Digital Arts and Experimental Media, University of Washington (Seattle, USA)

odiliscia@unq.edu.ar, pampin@u.washington.edu

Abstract. *As a way of improving the results of the FFT based analysis and synthesis of digital sound, some researchers applied high resolution analysis combined with a model which attempts to represent the attributes of a sound taking in account its deterministic and stochastic parts(see Serra and Smith, 1990, and Pampin, 1999).*

This paper deals both with the ATS (analysis, transformation and synthesis) technique and with a graphic application developed to handle it, the ATSH program.

1-General Background

From the very beginning of computer music, the *Fast Fourier Transform* (FFT) has been considered a powerful technique that allowed the development of many valuable tools for research and transformation of digital sound. The reader may consult -among others- the works by (Moore,1990, 1978, Embree & Kimble, 1991, Moorer, 1978, and Wessel & Risset, 1985), in order to obtain the basic concepts needed for a detailed comprehension of the following discussion.

As a way of improving the results of the FFT based analysis and synthesis of digital sound, some researchers applied *high resolution analysis* combined with a model which attempts to represent the attributes of a sound taking in account its *deterministic* and *stochastic* parts(see Serra & Smith, 1990, and García & Pampin, 1999). Generally speaking, the latter method implies that the data obtained using the short-term FFT analysis is further refined and encoded as time varying sinusoidal trajectories (representing the *deterministic* part of the signal) by one hand, and as spectrally changing noise (representing the *residual*, or *stochastic* part of the signal) by the other. This allows both a data representation closely related to musical experience, and more accurate results on the transforming/synthesizing processes.

2-About the ATS System

ATS (Pampin, 1999) is a system for sound Analysis, Transformation, and Synthesis based on a sinusoidal plus critical-band noise model and psychoacoustic information.

In the Analysis part of ATS, partials are tracked using high-resolution sinusoidal analysis. Tracked partials are then synthesized using interpolated phase information, and subtracted from the analyzed sound in the time domain to obtain a residual signal. This residual signal contains data that couldn't be modeled by sinusoidal analysis (i.e. it is normally a noise-based signal). The resulting residual is modeled as time-varying critical-band noise. This is performed by warping the frequency spectrum of

the residual into a Bark scale, dividing it into 25 critical bands, and computing the energy in each band at the frame rate. Critical-band energy is then re-injected into partials present in those spectral regions as modulated narrow-band noise. In this way, sinusoidal and noise information are encapsulated into one data abstraction called a partial (a "noisy" sinusoid). One of the advantages of this encapsulated data representation is that sinusoidal and noise components are perceptually well integrated when synthesized. If perceptually relevant noise energy information is present within critical-band regions where no partials were tracked, a complementary noise model is used to keep it.

As part of the analysis process, psychoacoustic information is used to measure the perceptual relevance of detected spectral peaks. This information (measured as signal-to-mask ratio, or SMR) is derived from masking effects produced within critical bands, and accounts for the audibility of sinusoidal trajectories at a particular analysis frame or across a time window. To achieve coherent sinusoidal trajectories, both SMR and frequency deviation information are used to track partials across frames. SMR information is also used as a psychoacoustic metric in the Transformation and Synthesis parts of the system. For instance, decisions about which partials to select for transformation can be made given a SMR threshold or, in case of limited synthesis resources, this information can be used to select a subset of "relevant" partials to synthesize, losing as little perceptual quality as possible.

Data issued by the analysis part of ATS represents a composite spectral model containing sinusoidal and noise information, and is stored in a binary file format which will be described in what follows.

3-The ATS binary files data structure

Generally speaking, the ATS files hold a representation of a digital sound signal in terms of sinusoidal trajectories (called *partials*) with instantaneous frequency, amplitude, and phase changing along temporal frames. Each frame has a set of partials, each of which having (at least) amplitude and frequency values (phase information might be discarded from the analysis). Each frame might also contain noise information, modeled as time-varying energy in the 25 critical bands of the analysis residual.

The ATS files starts with a header at which several data is stored. The table shown below displays the header structure together with a brief explanation of each data:

Data type	Meaning
64 bits double	Magic Number for ID of file (must be always 123.00)
64 bits double	Sampling Rate in Hertz
64 bits double	Frame size (in samples)
64 bits double	Analysis Window size(in samples)
64 bits double	Number of partials on each frame
64 bits double	Number of frames
64 bits double	Maximal amplitude value found.
64 bits double	Maximal frequency value found.
64 bits double	Duration (in seconds)
64 bits double	File Type: 1 =only amplitude and frequency data on file. 2 =amplitude, frequency and phase data on file. 3 =amplitude, frequency and residual data on file. 4 =amplitude, frequency, phase and residual data on file.

After the header data, the time, amplitude, frequency, phase and residual (these two latter may or may not be present) data of each partial in each frame is stored as a 64 bits double value.

4-About ATSH

ATSH is a program for analysis, transformation, and synthesis of digital sound by means of the ATS system. The objective of ATSH is to allow the using of the ATS system through a Graphic Interface. It is being developed by Oscar Pablo Di Liscia (Universidad Nacional de Quilmes, Argentina), Juan Pampin(Washington University, Seattle, USA), and Pete Moss(Washington University, Seattle, USA).

ATSH was originally developed to be run under Linux, using the X windows system. The source

code was written in the C programming language using GTK-GDK 1.2.0. It was compiled and tested successfully using several of the most popular Linux distributions (such as Red Hat, Debian, Mandrake, etc.). Installation should be straight forward if updated versions of Linux and the Gnome, GTK, and GDK libraries are properly installed in the user's computer.

It is possible also to compile and run ATSH under Apple Macintosh computers using the OS10 Operating System, and under Microsoft Windows if the required libraries (WinGtk) are properly installed.

At present, ATSH is a sort of viewer/editor of the analysis files generated by the ATS system (binary files usually carrying the *.ats extension).

5-The ATSH data display

The following picture shows a snapshot of the main window of ATSH:

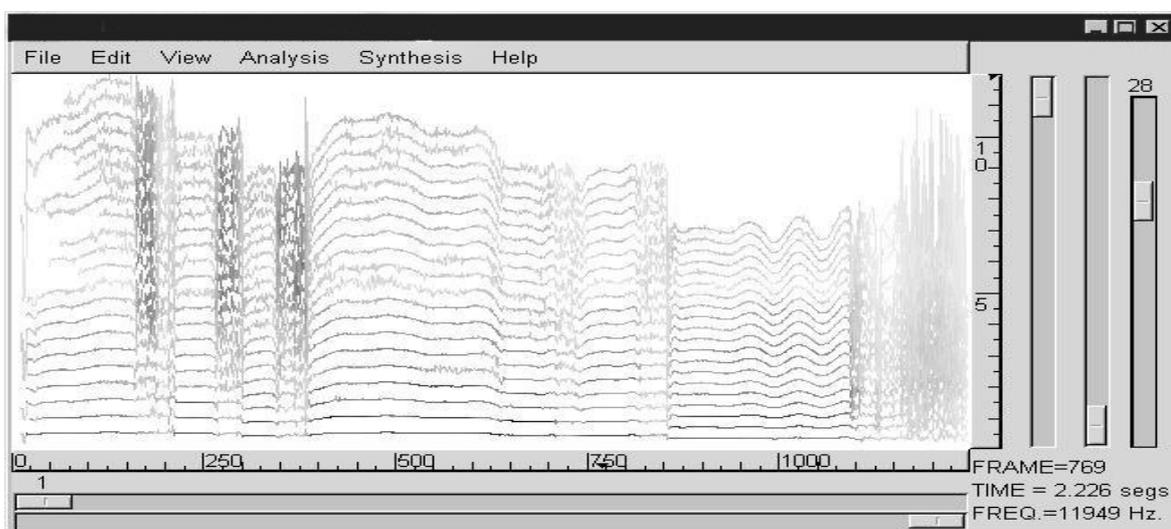


Figure 1: a screenshot of the ATSH program main window.

It can be seen that the frequency of each partial is represented on the vertical (Y) axis, Time (in frames) runs along the horizontal (X) axis, and amplitude is represented with a color value.

The two horizontal scrollbars control the time (frame) view. The top one controls the from-view value, and the bottom one controls the size of the view.

There are three vertical scrollbars as well. The two left-most ones control the frequency view (in a similar way the horizontal scrollbars control the time view), and the right-most scrollbar controls a contrast value for the amplitude display.

Horizontal and vertical scrollbars can be used to select and zoom in/out zones of the spectral data. The contrast slider adjusts partials amplitude display: a value of 50 shows the *normal* contrast between loud and quiet partials, while a value of 100 overrides amplitude information (i.e. all partials are displayed black). A value of 0 shows only very loud partials.

It is also possible to see the value of each of the data at the file's header choosing */view/file header*.

6-Selecting data

To make any changes, the user must select some data. ATSH performs both, a horizontal (frame) and a vertical (partial) selection. There are at present four ways of spectral data selection:

1-Using some presets from the *Edit* menu. There are *Select All*, *Unselect All*, *Select Even*, *Select Odd*, and *Invert Selection* routines.

2-Clicking with the mouse at the graphic screen.

3-Using the *List View* window (menu *View*). In this menu all the data can be seen under the form of a numerical list. The amplitude, frequency and phase (if any) values of each frame are represented at each page of the list and may be selected or deselected individually or by groups.

4-Using the *smart selection* window: the user may specify a frequency step value and an amplitude threshold in order to select partials whose (peak or RMS) amplitude values are above the threshold.

7-Analyzing Digital Sound

The ATSH program allows the user either to load an *ats* analysis file or to create one by analyzing a sound file. In the *File/New ATS File* menu the user may enter the parameters for the analysis. The result of the analysis is held in memory until the user quit the program or save it on a file.

8-Transforming the selected data

At present there are three ways to transform the selected data: the *Edit/Amplitude* menu, the *Edit/Frequency* menu, and the *Synthesis/Set time pointer* menu.

The *Edit/Amplitude* menu and the *Edit/Frequency* menu allows the user to draw a function that will be applied to the amplitude or to the frequency values of the partials over the selected time region. The frequency and amplitude values of the partials selected may be either scaled (multiplied by the function), or "shifted" (have the function values added to them). The functions may be either linear or spline shaped. There is an "unlimited" *Undo* choice for the editions.

Straight forward time-varying filtering effects may be obtained very easily performing amplitude changes over different kind of selection as this allows the user dynamic weighting of the amplitude of each partial or a bunch of them over a time span.

By the other way, the frequency envelope allows many interesting and smooth transformations of spectral quality as the frequency of each partial or a bunch of them may be dynamically modified over a time span.

The *Synthesis/Set time pointer* menu is explained below.

9-Synthesizing

At present, ATSH's synthesis engine is implemented as an array of linear interpolating table-lookup oscillators. The program writes the result of the synthesis on a soundfile which may have the WAVE, AIF or SND format.

Several features concerning synthesis may be set on the *Synthesis/Parameters* menu. The user may scale the overall amplitude and frequency of the original data using scalars. Note also that synthesis may use all the data, or just a selection (if any).

10-Data timing transformation

As is very characteristic on this kind of synthesis, the timing of the data may be transformed on an independent way of its other attributes. This is done graphically on a similar fashion with which most spectral synthesis unit generator works (See Karpen, 1998). By setting up a time function the user may stretch or expand the file data dynamically, as well as read it forward or backwards. The duration of the output file is represented on the X (horizontal) axis while the temporal location of the data of the analysis to be used in the synthesis is represented on the Y (vertical) axis. The slope of the line at each segment will produce stretching (sharp slope), expansion (non sharp slope) or time invariance ("normal" slope). Rising slopes will produce forward synthesis whilst falling slopes will produce backward synthesis.

11-Conclusions and future improvements

The software ATSH has proved to be an interesting tool for analysis and transformation of digital sound. The synthesis resources will be improved including also subtractive synthesis by means of a bank of resonant filters in parallel connection each one with variable resonant frequency and bandwidth adjusted

to the corresponding frequency and amplitude of each partial. The transforming and editing resources will be further improved including spectral morphing and intelligent selection algorithms.

12-Acknowledgements

To the *Center for Digital Arts and Experimental Media*(University of Washington, Seattle, USA) and *Universidad Nacional de Quilmes*(Buenos Aires, Argentina) for supporting the Research Project(*Software development for digital sound analysis and synthesis*) that made possible the development of ATSH.

To the GTK Developer Team, for the *GTK* toolkit which allowed to program the GUI as well as porting it to different OS.

To Bill Schottstaedt(CCRMA, Stanford University) for the C language library *Sndlib* which was used in the Audio Files I/O.

13-References

Embree, P. & Kimble, B. (1991): *C language algorithms for DSP*, Prentice Hall, New Jersey, USA.

Karpen, R.(1998): *Phase Vocoder Resynthesis* In: *The Csound Manual*, (<http://hem.passagen.se/rasmuse/Csound.htm>), MIT.

Moore, F. R.(1978): *An introduction to the mathematics of DSP, Part II*, CMJ 2(2):38-60, MIT Press, USA.

Moore, F.R.(1990): *Elements of Computer Music*. Prentice Hall., New Jersey.

García, G. and Pampin, J.(1999): *Data compression of sinusoidal modeling parameters based on psychoacoustic masking*, in Proc. of the Int. Computer Music Conference, Beijing.

Pampin, J. (1999): *ATS: a Lisp environment for Spectral Modeling*, in Proc. of the Int. Computer Music Conference, Beijing.

Pampin, J. (1999-2003): <http://www-ccrma.stanford.edu/~juan/ATS.html>.

Serra, X. and Smith J. O. III (1990): *A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition*, Computer Music Journal, Vol.14 #4, MIT Press, USA.

Moorer, J. A. (1978): *The use of the Phase Vocoder in Computer Music Applications*, JAES, 26(1/2): 42-45.

Wessel, D. and Risset, J. (1985): *Exploration of Timbre by Analysis and Resynthesis*, pp. 26-58 in *The Psychology of Music*, ed. D. Deutsch, Academic Press.