InSpect and ReSpect: spectral modeling, analysis and real-time synthesis software tools for researchers and composers

Sylvain Marchand (`sm@labri.u-bordeaux.fr`)
Robert Strandh (`strandh@labri.u-bordeaux.fr`)

SCRIME - LaBRI - Université Bordeaux I
351 cours de la Libération, F-33405 Talence Cedex, France

Abstract

We describe a software system for analysis, manipulation, and synthesis of low-noise sounds. The system is based on a model derived from additive synthesis, and opens new horizons for both researchers and composers. The software system consists of two separate tools: InSpect and ReSpect. InSpect performs the analysis of sampled sounds, extracting parameters and structuring them into its spectral sound model. The resulting spectral sounds can be manipulated in a very musical and intuitive way. We provide traditional operations such as filtering, time stretching, cross-synthesis, transposition while preserving formants, timbre morphing and many more. ReSpect is able to re-synthesize the audio signal from its spectral representation in real time.

1 Introduction

In order to faithfully imitate and also transform existing sounds using a computer, a formal representation is needed for these sounds. Spectral models provide general representations in which such operations can be performed in a very natural and musically expressive way, provided that an accurate analysis method is able to extract parameters from the sounds.

Many analysis methods are available but few were implemented in a software system, and it appears that freely-available analysis software tools are extremely rare. We present in this paper a software system for both the Linux and the Windows 95/98/NT operating systems that is distributed freely. It is composed of two software tools written in the C programming language: InSpect and ReSpect.

InSpect is presented in section 2. It was developed in order to convert sounds from the temporal model (audio signal amplitude versus time) to a spectral representation, allowing intuitive and musical manipulations. Although it was first designed for low-noise sounds, it can also analyze, transform, and re-synthesize noises as well. But InSpect does not yet separate the sinusoids (deterministic part) from the noise (stochastic part) like SMS [Ser97] does.

ReSpect is presented in section 3. This software tool was designed in order to perform real-time synthesis of sounds from their spectral representation.

2 InSpect: Spectral Analysis

InSpect ("Inspect Spectrum") [Mar98b] is a sound analysis program, designed to look at the inner structures of sounds. It performs the analysis of sampled sounds, extracting parameters and structuring them into a spectral sound model derived from additive synthesis. The resulting spectral sounds can be then manipulated in a very musical and intuitive way. Although InSpect can re-synthesize these sounds, ReSpect was specially designed for the purpose of real-time synthesis.

2.1 Sound Model

InSpect is based on additive synthesis, in which a sound is represented as a sum of oscillations. Each oscillation is produced by a sinusoidal oscillator which frequency and amplitude vary slowly with time. Such an oscillator is commonly called a *partial*. The audio signal a can be calculated using the following equations:

$$a(t) = \sum_{p=1}^{P} a_p(t)\ sin(\phi_p(t)) \tag{1}$$

$$\phi_p(t) = \phi_p(0) + 2\pi \int_0^t f_p(u)\ du \tag{2}$$

where P is the number of partials and f_p, a_p, and ϕ_p are respectively the instantaneous frequency, amplitude and phase of the p-ieth partial.

InSpect can also structure these additive parameters in order to switch to the Structured Additive Synthesis model (SAS) [DCM99], well-suited for intuitive and musical transformations.

But since this functionality is still experimental in the actual version of the software tool, we won't develop it further in this presentation.

2.2 Overview

Basically, InSpect can open a sound file, analyze it, and synthesize a new one from the analysis. The synthesized sound file may be played or exported. InSpect does neither care about recording, nor about the way the original sound file was obtained. Neither does it compare the resulting sounds, which is strictly the role of the (human) user. Figure 1 gives a general overview of the main functionalities of InSpect. InSpect allows you to look at the inner structures of sounds. For each partial you can show its frequency and amplitude as functions of time. You can also display the short-time spectrum at a given time and the associated spectral envelope, as well as spectrograms, phasograms, and results of linear prediction…

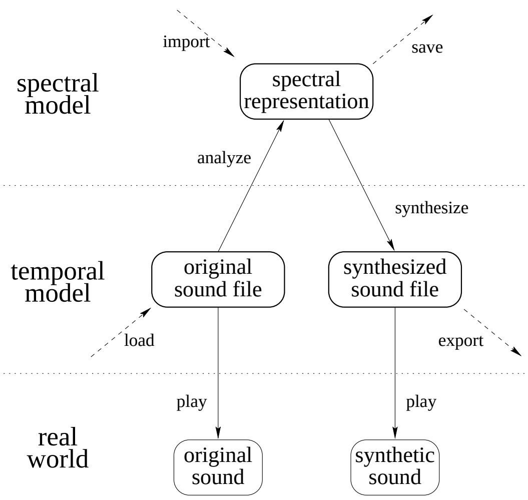

Figure 1: InSpect Architecture

2.2.1 Step 1: Loading

The first step is to load a sampled sound. InSpect supports many file formats. Traditional computer sound files (WAV, AIFF, etc.) are typical examples of representations of sounds in the temporal model.

2.2.2 Step 2: Analysis

InSpect was designed not only for researchers and engineers, but also for composers and musicians. That's why the analysis step can be performed without setting lots of parameters. The most useful analysis configurations are available instead, according to the nature and the pitch of the sound to analyze. However the default configuration succeeds in most cases. The default method of InSpect works best with low-noise sounds, harmonic or not. The analysis step and its configuration are developed later in this section.

2.2.3 Step 3: Saving

When the analysis step is completed, the resulting spectral sound is displayed as show in Figure 2 and you can save it in a spectral format. InSpect knows how to take advantage of the slow time-varying nature of the parameters in order to perform efficient compression. It is also possible to import directly an already-analyzed sound, thus skipping the analysis step.

2.3 Analysis

Given a sampled sound, the aim of the analysis phase is to decompose it into elementary sound components called partials, whose frequencies and amplitudes evolve slowly in time. This is the well-known McAulay-Quatieri analysis [MQ86] used in Lemur [FH96] and SMS [Ser97]. InSpect performs this analysis in three steps.

2.3.1 Step 1: Short-Time Analysis

Our analysis tool uses a new technique [Mar98a] for extracting parameters from sampled sounds. In addition to the signal itself this new technique uses the derivative of the signal. We compute the discrete Fourier transform (using the FFT algorithm) of both the signal and the derivative of the signal. The quotients between peaks across these two spectra allow us to compute very accurate values for the instantaneous frequencies of the sound, which are then used to precisely compute the corresponding amplitudes. Our method gives very accurate values for frequency and amplitude contents, provided that partials are separated by at least the lowest frequency of the Fourier transforms used. Experience has shown that our method can work with very short analysis windows, speeding up the analysis while maintaining a high-quality result together with an increased time resolution.

InSpect was first developed in order to experiment in practice this enhancement of the standard short-time Fourier analysis called the n-th order Fourier analysis [Mar98a]. However other analysis methods are implemented to make comparisons possible. Among the analysis parameters you can change are the analysis window width and type (Hanning, Kaiser, Bartlett, Hamming, Blackman, etc.).

2.3.2 Step 2: Connection

Each short-time analysis produces a spectral frame. Spectral peaks are tracked from frame to frame to form partials. Different connection strategies are available, as well as connection thresholds in frequency and amplitude.

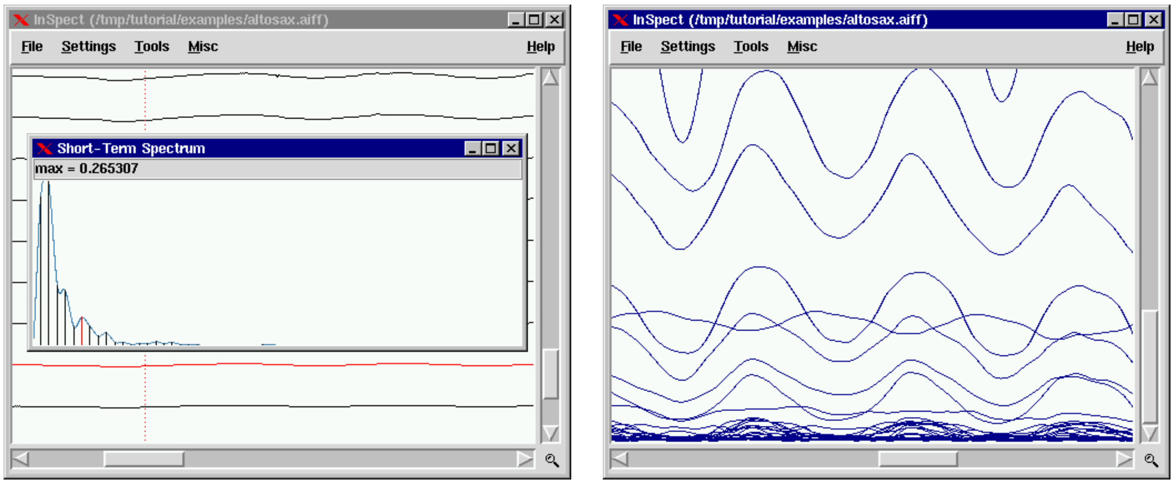

Figure 2: InSpect displaying the evolutions of the partials of an alto saxophone as functions of time during 0.7 seconds. The snapshot on the left shows the frequencies of partials, as well as a short-time spectrum and the corresponding spectral envelope, while the one on the right shows the amplitudes of partials.

2.3.3 Step 3: Selection

The result of the connection step is a (frequently large) set of partials. InSpect selects only some of these partials according to certain selection methods and criteria. The default method is selecting the strongest and longest partials, given thresholds in strength and duration.

2.4 Transformations

The result of the analysis is a set of partials whose parameters (frequency and amplitude) evolve relatively slowly with time. Our software system allows us to structure and manipulate this intermediate representation in a very musical and intuitive way. We provide musical parameter extraction (volume, pitch, brightness) as well as traditional operations such as filtering, time stretching (without limits on stretching factor), cross-synthesis, transposition while preserving formants, timbre morphing and many more. Since these manipulations are made on a spectral representation which has fewer data points than the signal itself, our operations are quite efficient.

2.5 Re-synthesis

From the spectral representation, it is then possible to re-synthesize a sound back in the temporal model. By playing the original and re-synthesized sounds and comparing them, one can hear if the original sound has been faithfully modeled. Again different synthesis methods are proposed, mainly for research and experimentation purposes. The fastest synthesis method is available in a separate software tool called ReSpect.

3 ReSpect: Fast Synthesis

ReSpect ("Respect Spectrum") is a sound synthesis program performing the real-time synthesis of spectral sounds. It can generate many oscillators simultaneously and behaves as a virtual sound card accepting spectral sounds.

3.1 Software Oscillators

Although other methods have been proposed [FRD93], we chose to generate each oscillator using a simple recursive description [SC92] which is, for a discrete signal with sampling period T:

$$s[i+1] = 2cos(2\pi fT) \cdot s[i] - s[i-1] \qquad (3)$$

This method is indeed more flexible and very efficient, as shown in [SM99]. This method allows an extremely fine control of each oscillator, and we compute a new sample with one floating-point multiplication and one floating-point addition, which on most platforms comes to only a few clock cycles. Numeric stability is not a problem since we measure and adjust the parameters regularly and often enough to avoid any drift, and rarely enough to preserve performance. Currently, we obtain about 2 oscillators per MHz of clock frequency on Intel Pentium II processors, so that a 400 MHz processor gives us 800 oscillators in real time. Although we have not investigated this yet, we believe that we can use results from psychoacoustics to show that masking and other phenomena will allow us to produce any sound with less than 1000 oscillators or so. As a consequence, in a few years most people will have a PC with sufficient power on their desks to generate any sound in real time based on our method.

3.2 Virtual Sound-cards

When ReSpect is installed as a module in the UNIX kernel, the re-synthesis is controlled by by writing in a special device driver: `/dev/respect`. This driver maintains an ordered list of the active partials currently synthesized. Each partial p is represented by a (f_p, a_p) pair of floating-point numbers in double precision. Recall that f_p and a_p are respectively the frequency and amplitude of the p-ieth partial. The number of synthesized partials can vary with time. When a partial dies, the special $(0, 0)$ pair is placed at its position in the partials list. After that, the partial does not exist anymore. When a new partial appears, its is simply appended to the partials list. When the pairs are written into the device driver, the following protocol must be respected in sequence:

1. For each partial p, write its (f_p, a_p) pair. If a partial dies, the $(0, 0)$ pair is written.
2. For each new partial, write its pair.
3. Write the $(-1, -1)$ end-marker pair.

This steps should be repeated for each temporal frame, at a frequency depending on the sampling period. Since we are in the spectral model, this period can be as low as 40 samples per second.

Since ReSpect has a direct hardware access, we can guarantee that there will be no click during a live performance using ReSpect. In the worst case we will hear a steady sound for a few milliseconds if the spectral information does not arrive on time at the device driver.

4 Conclusion

We have described in this paper a software system for analysis, manipulation, and synthesis of spectral sounds. InSpect is well-suited for analysis purposes, while ReSpect performs real-time re-synthesis.

While we will keep on optimizing ReSpect and its synthesis algorithm, the development of InSpect is stopped. InSpect was first designed for analysis sound only. In order to perform musical sound transformations, we are now instead developing a new research tool also useful for creation. This tool will combine the advantages of the C programming language with the power of functional languages (Scheme or Lisp). The development version of this tool has been tested by composers of electro-acoustic music who were enthusiastic. This forthcoming software system should open new horizons for both researchers and composers. It will be freely distributed under the terms of the GNU Public License.

References

[DCM99] Myriam Desainte-Catherine and Sylvain Marchand. Structured Additive Synthesis: Towards a Model of Sound Timbre and Electroacoustic Music Forms. In *Proceedings of the International Computer Music Conference (ICMC'99, Beijing)*, 1999.

[FH96] Kelly Fitz and Lippold Haken. Sinusoidal Modeling and Manipulation Using Lemur. *Computer Music Journal*, 20(4):44–59, 1996.

[FRD93] Adrian Freed, Xavier Rodet, and Philippe Depalle. Synthesis and Control of Hundreds of Sinusoidal Partials on a Desktop Computer without Custom Hardware. In *Proceedings of the International Computer Music Conference (ICMC'93, Tokyo)*, 1993.

[Mar98a] Sylvain Marchand. Improving Spectral Analysis Precision with an Enhanced Phase Vocoder using Signal Derivatives. In *Proceedings of the Digital Audio Effects Workshop (DAFX'98, Barcelona)*, pages 114–118, 1998.

[Mar98b] Sylvain Marchand. InSpect Software Package. URL http://www.scrime.u-bordeaux.fr, 1998.

[MQ86] Robert J. McAulay and Thomas F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, 1986.

[SC92] Julius O. Smith and Perry R. Cook. The Second-Order Digital Waveguide Oscillator. In *Proceedings of the International Computer Music Conference (ICMC'92, San Jose)*, pages 150–153, 1992.

[Ser97] Xavier Serra. *Musical Signal Processing*, chapter Musical Sound Modeling with Sinusoids plus Noise, pages 91–122. Studies on New Music Research. Swets & Zeitlinger, Lisse, the Netherlands, 1997.

[SM99] Robert Strandh and Sylvain Marchand. Real-Time Generation of Sound from Parameters of Additive Synthesis. In *Proceedings of the Journées d'Informatique Musicale (JIM'99)*, pages 83–88, 1999.