AudioGaming

NATURAL BORN INTERACTIVE

INP TOULOUSE ENSEEIHT

Long Project with Audiogaming

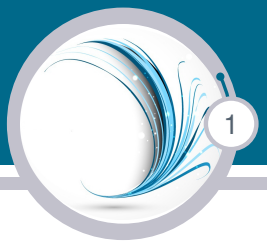
Additive Synthesis with Inverse Fourier Transform for Non-Stationary Signals

Clément Cazorla - Vincent Chrun - Bastien Fundaro - Clément Maliet

Audiogaming Supervisor : Chunghsin Yeh

March 03, 2017

Content



AudioGaming

NATURAL BORN INTERACTIVE

- Localization: Toulouse, Paris
- Activity: Audio plug-in (VSTs and RTAS)
- Main customers: Film and Video Game Industry (Sony, Ubisoft)
- 10 employees

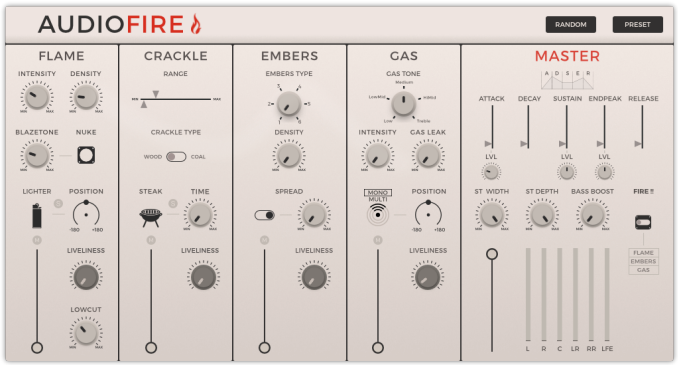
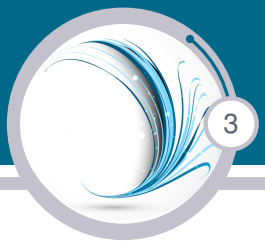


Figure: Audiofire: audio plug-in that recreates fire sound



- We are continuing the Audiogaming long project from 2015 (Emilie Abia, Lili Zheng, Quentin Biache)

Objective : Synthesizing sounds from their spectrum with a FFT^{-1}

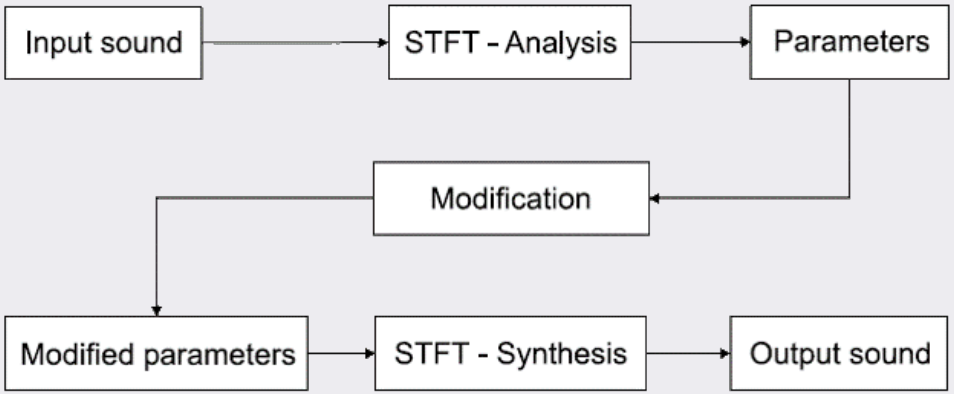
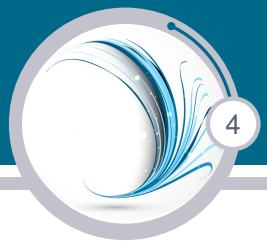


Figure: General approach for modifying a sound in the spectral domain

- We have to implement a new method of additive synthesis $\Rightarrow$ computationally very fast



- 6 weeks only $\Rightarrow$ Focus on the synthesis method only.

Given codes in Python and Matlab from the 2015 project :

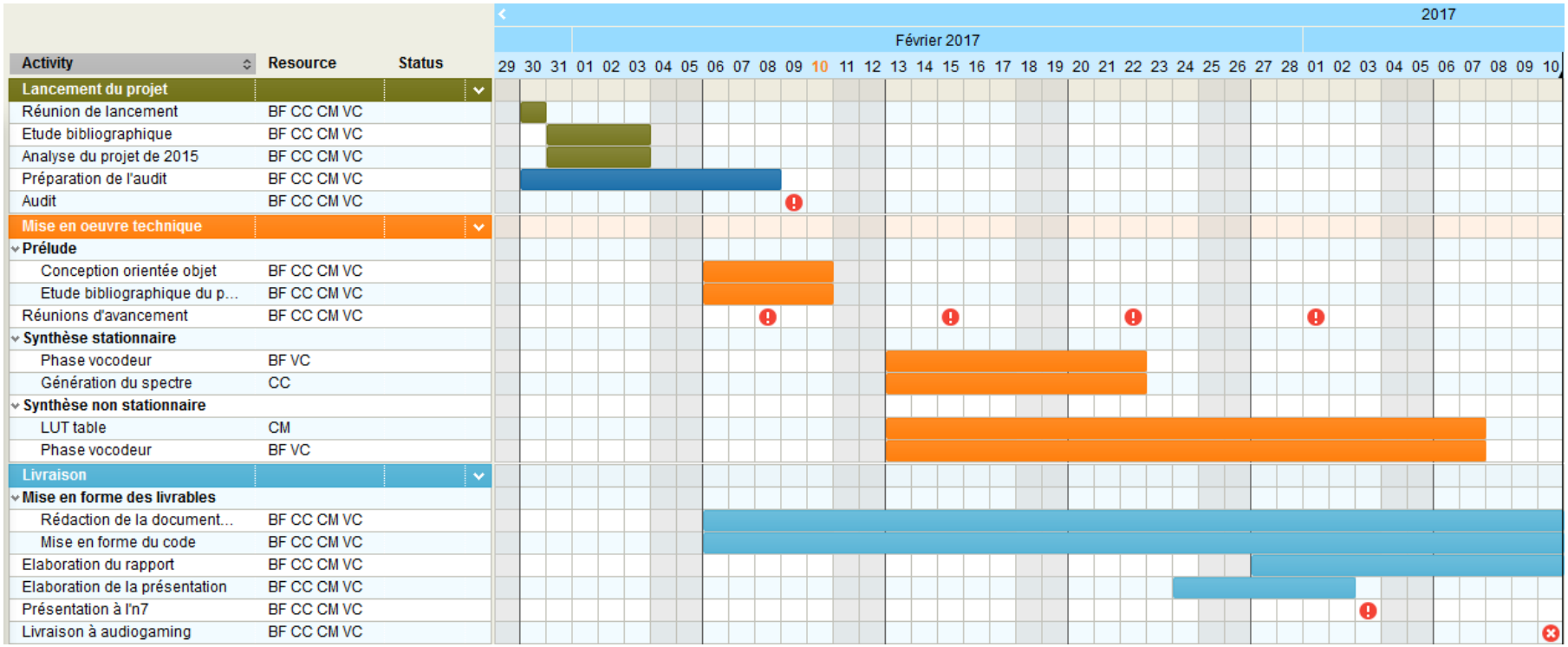
- Python : Analysis estimator of sinus parameters and sinus generation with those parameters (only stationary)
- Matlab : Some reasearch on the Non-stationary synthesis with the LUT of lobes

- We made our own OOP codes in Python
- We have taken the analysis estimator code to test our final synthesis

Figure: *PyCharm* as Python IDE , *Slack* to communicate,*GitHub* to stock the codes and have a versionning, *Freedcamp* to plan the project events

Introduction

Project Management : Gantt Chart



Method Overview : Analysis
Windowing

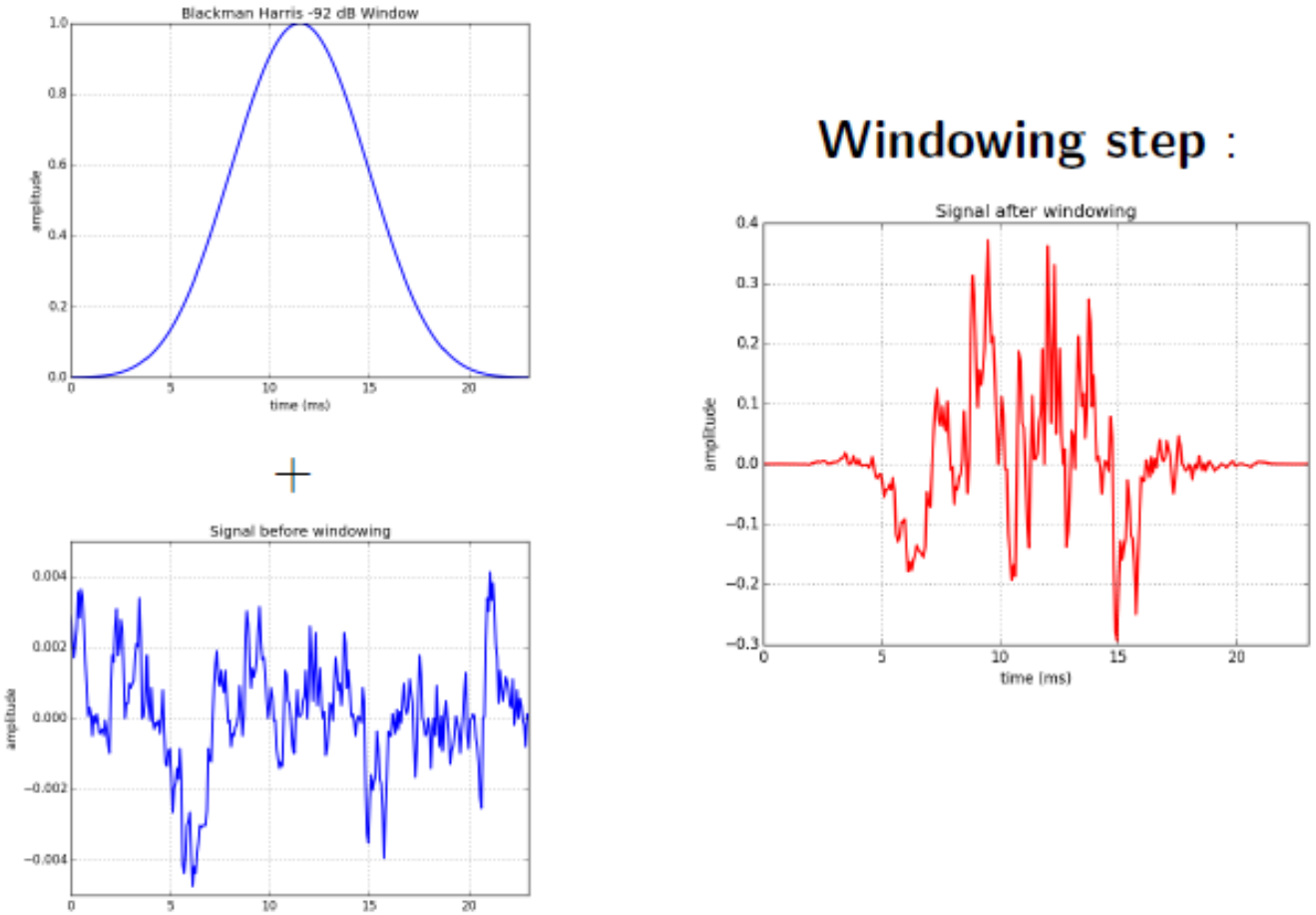


Figure: *Windowing step*

Method Overview : Analysis

Peak Detection

Peak detection and extraction of parameters by STPT (particular Short Time Fourier Transform):

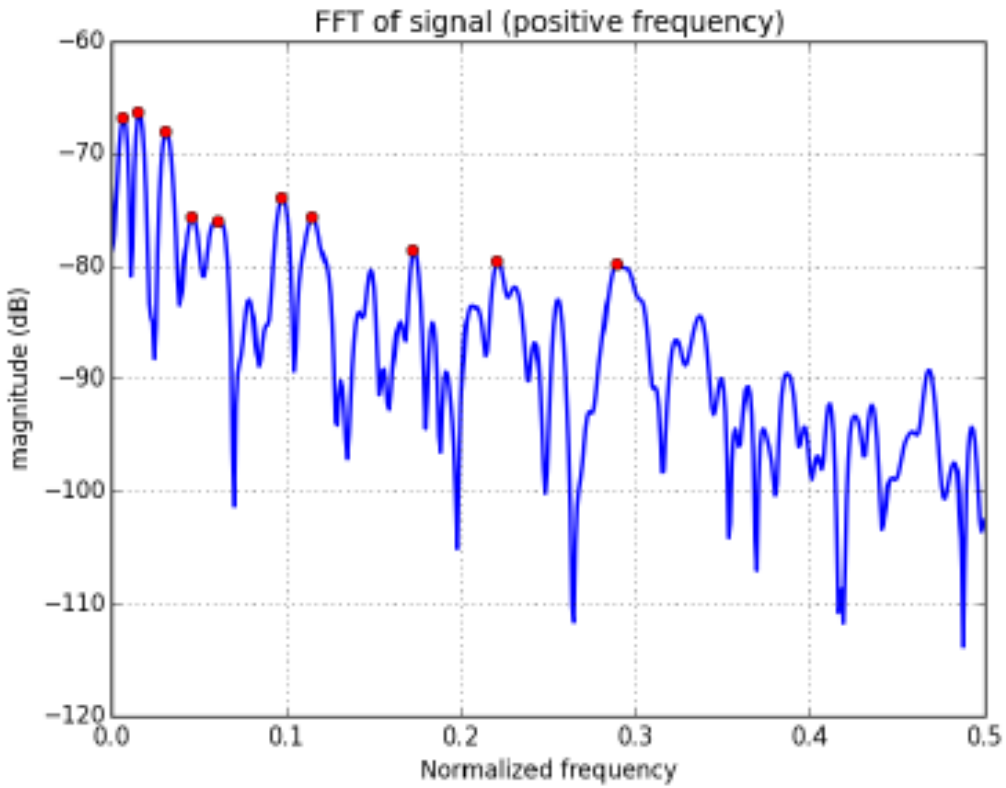


Figure: *Peak detection*

Method Overview : Synthesis

Result

Additive synthesis according to the parameters from the analysis:

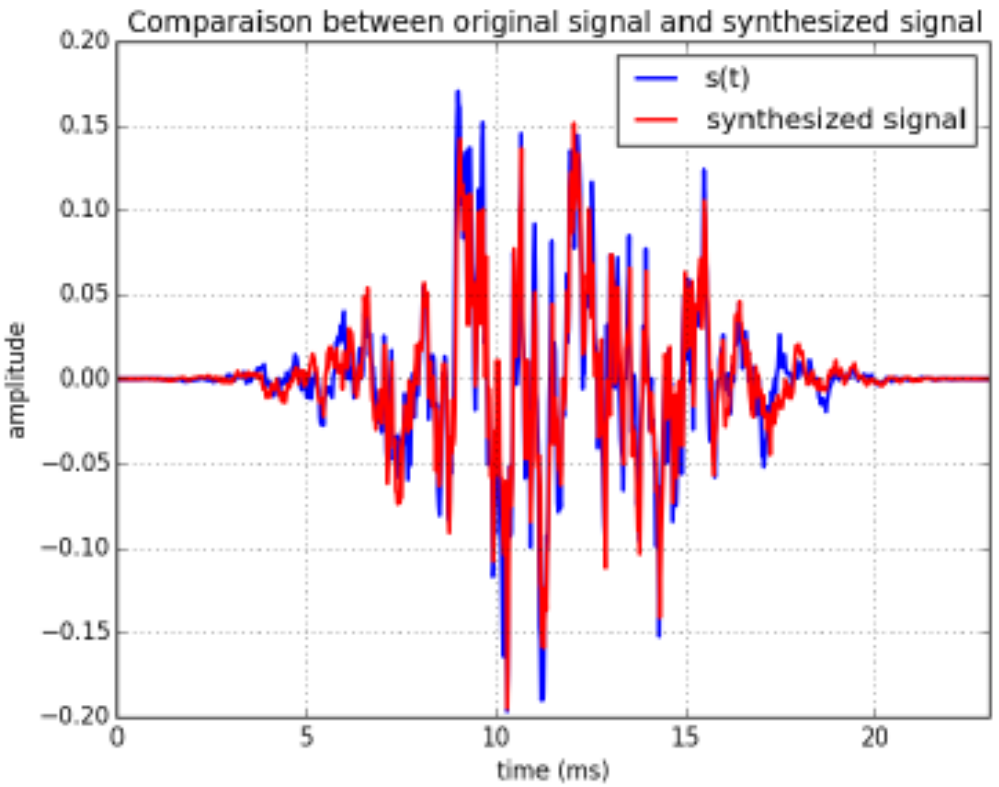


Figure: *Synthesized signal vs Original signal*

The additive synthesis
General approach: The time domain

The sound signal is represented as a sum of N sinusoids:

$$x(t) = \sum_{n=1}^{N} a_n sin(2\pi f_n t + \phi_n)$$

- Very costly to implement
- Impossible to compute in real-time

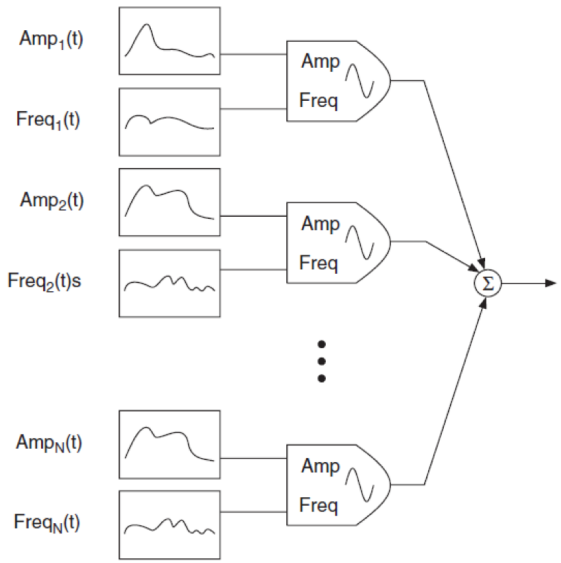


Figure: *The additive synthesis*

The additive synthesis

General approach: The frequency domain

We generate the sinusoids in frequency domain in order to reduce the computation time :

- Window the signal to maximize the energy in the main lobe
- We only keep the main lobe for each sine (9 points)
- We assume that the parameters (amplitude, frequency, phase) are already given by the analysis

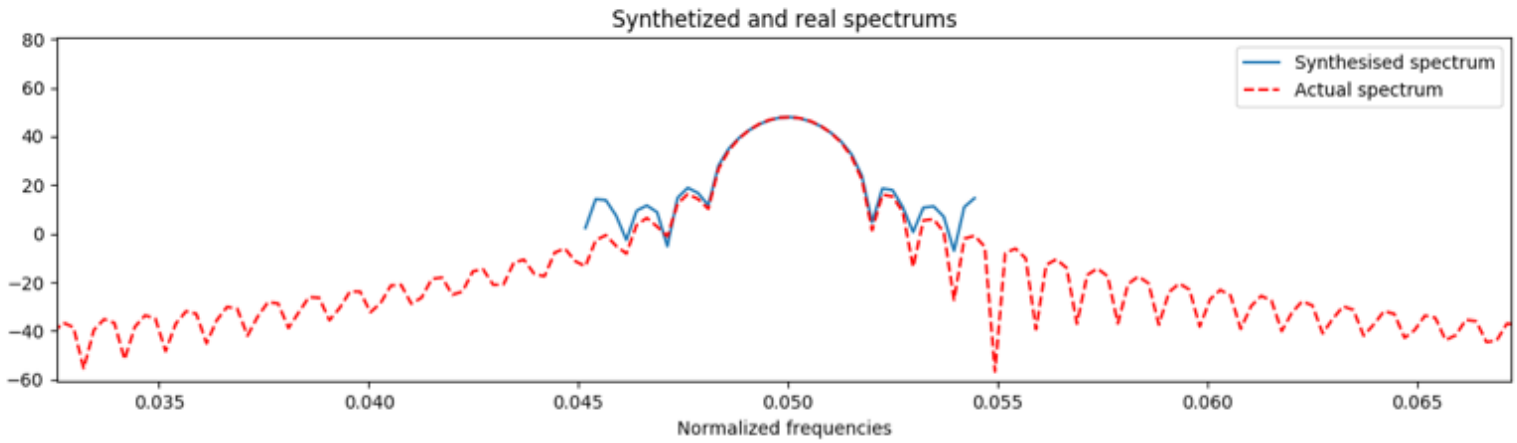


Figure: *Windowed sine lobe*

The additive synthesis

The frames

The sound signal is a frame-by-frame signal:

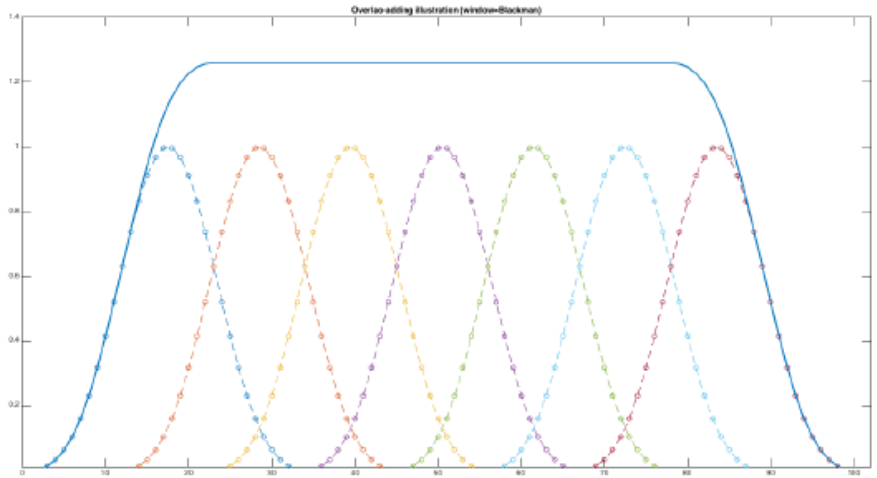


Figure: *Sum of small size Hanning windows*

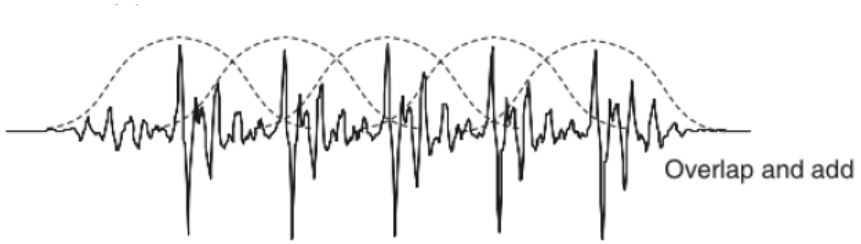


Figure: *Overlap and add*

Stationary Signals
Source files

-
-
-
-

n

-
-

-
-

-
-

–

–

–

- ▶
- ▶

-
-

Thank you!