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

- Introduction
  - The company
  - Objective
  - Context of the Project
  - Work Environment and Project Management
- Method Overview
  - Additive Synthesis (Time Domain)
  - Windowing
  - Peak Detection
  - Result
- The additive synthesis
  - General approach: The time domain
  - General approach: The frequency domain
  - General approach: The frames
- Stationary Signals
  - hjh
- -
  - -

Introduction
The company

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

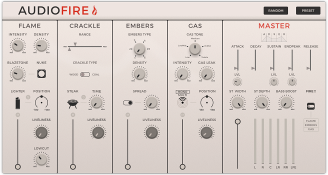


Figure: Audiofire: audio plug-in that recreates fire sound

Introduction

Objective

- 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}

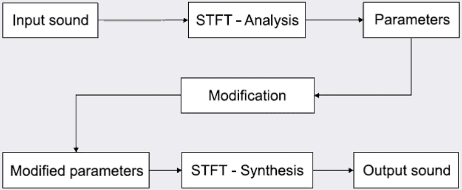


Figure: General approach for modifying a sound in the spectral domain

- We have to implement a new method of additive synthesis ⇒ computationally very fast

Introduction

Context of the Project

- 6 weeks only ⇒ 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

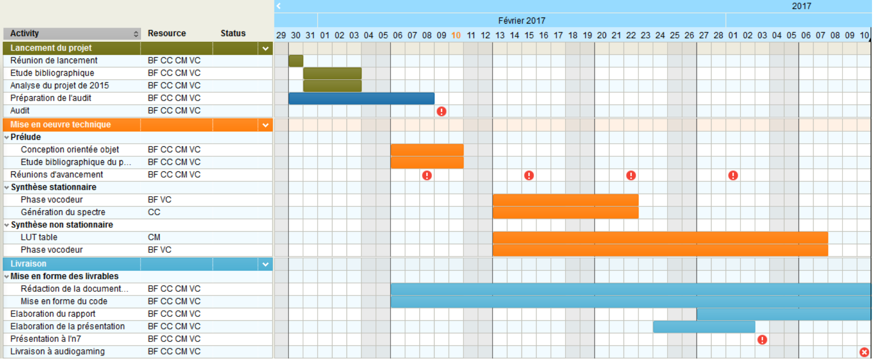
- We made our own OOP codes in Python
- We have taken the analysis estimator code to test our final synthesis

Introduction
Work Environment

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



Activity	Resource	Status
Lancement du projet		
Réunion de lancement	BF CC CM VC	
Etude bibliographique	BF CC CM VC	
Analyse du projet de 2015	BF CC CM VC	
Préparation de l'audit	BF CC CM VC	
Audit	BF CC CM VC	
Mise en oeuvre technique		
Prélude		
Conception orientée objet	BF CC CM VC	
Etude bibliographique du p...	BF CC CM VC	
Réunions d'avancement	BF CC CM VC	
Synthèse stationnaire		
Phase vocodeur	BF VC	
Génération du spectre	CC	
Synthèse non stationnaire		
LUT table	CM	
Phase vocodeur	BF VC	
Livraison		
Mise en forme des livrables		
Rédaction de la document...	BF CC CM VC	
Mise en forme du code	BF CC CM VC	
Elaboration du rapport	BF CC CM VC	
Elaboration de la présentation	BF CC CM VC	
Présentation à l'n7	BF CC CM VC	
Livraison à audiogaming	BF CC CM VC	

Method Overview : Additive synthesis

Additive Synthesis

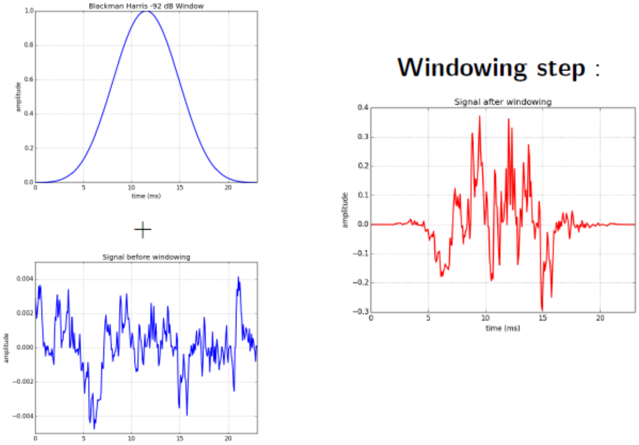
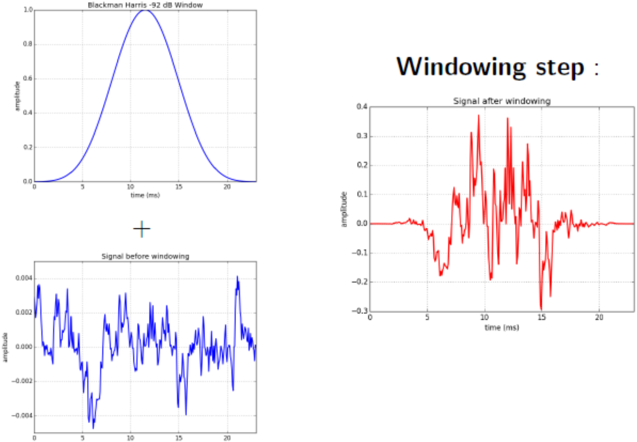


Figure: *Windowing step*

Method Overview : Analysis

Windowing



Windowing step :

Figure: *Windowing step*

Method Overview : Analysis

Peak Detection

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

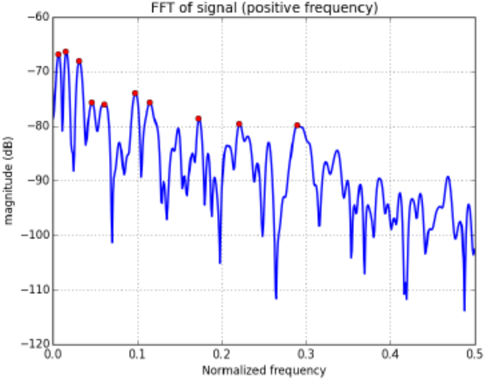


Figure: *Peak detection*

Method Overview : Synthesis

Result

Additive synthesis according to the parameters from the analysis:

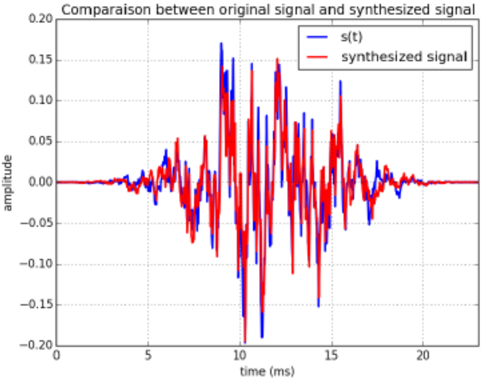


Figure: *Synthesized signal vs Original signal*

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

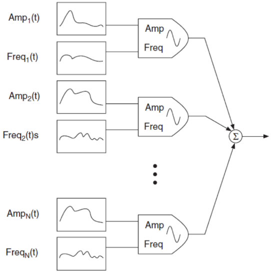


Figure: *The additive synthesis*

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

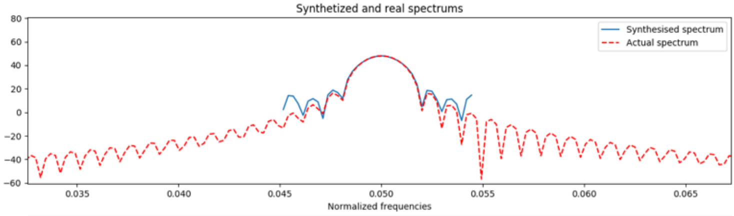


Figure: *Windowed sine lobe*

The additive synthesis

The frames

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

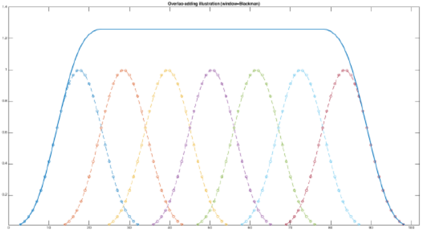


Figure: *Sum of small size Hanning windows*

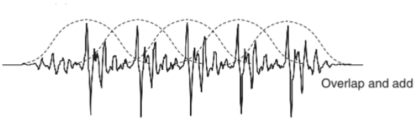


Figure: *Overlap and add*

Stationary Signals

Source files

-
-
-
-

n

Thank you!