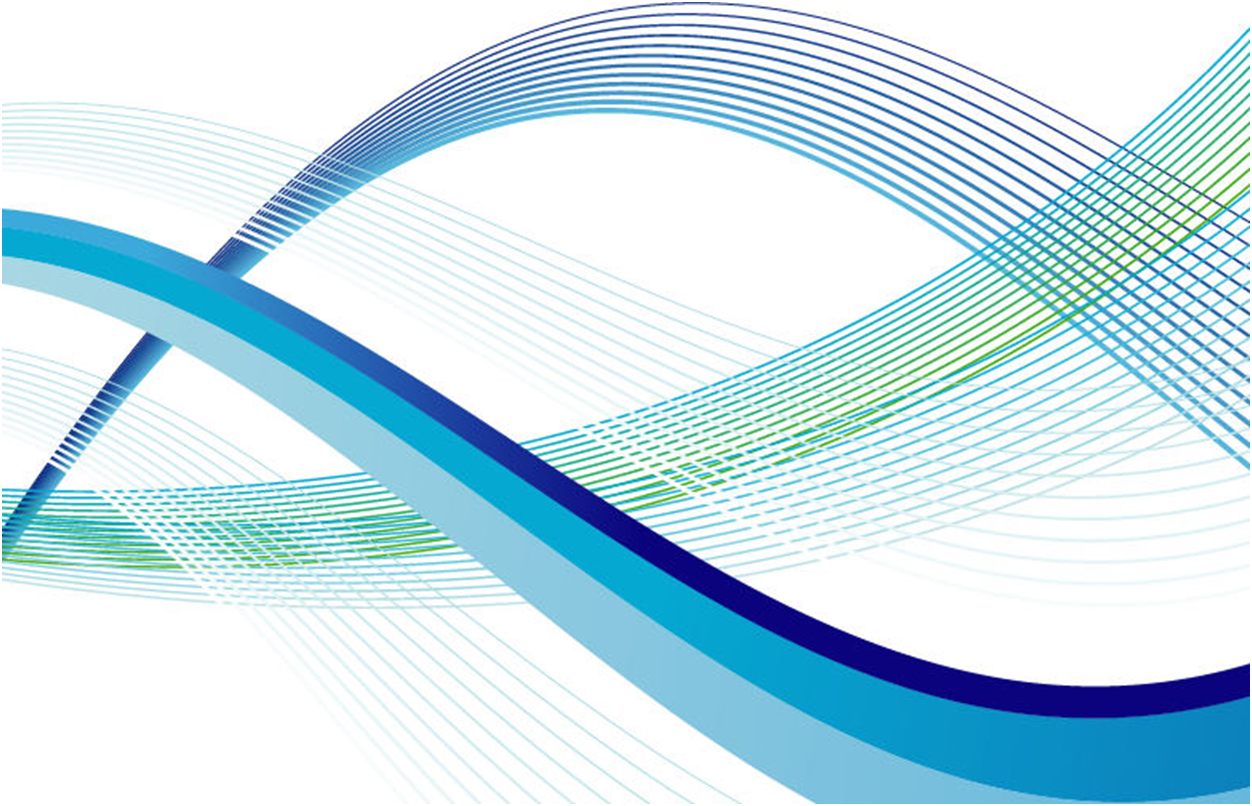
|  |
| --- |
| Polytech’ Nice Sophia Antipolis |
| Word recognition |
| Designing a basic model of word recognition |
|  |
| **Guénon Marie, Achard Jean-Paul, Favreau Jean-Dominique** |
| **24/1/2013** |

|  |
| --- |
|  |



Contents

[Introduction 3](#_Toc345400381)

[Plan prévu : 4](#_Toc345400382)

[Conclusion 5](#_Toc345400383)

# Introduction

# Intended plan:

* Sound recording (complete word)
* Sound treatment
  + Auto-correlation
  + Time split in slices of 20/30 ms, for all times
  + Fourier transformation for all slices 🡺 put together in a spectrogram (matrix)
* Learning (problem of Bakis / Hidden Markov/ Dynamic Time Warping)
  + Frequencies scaling: 20 significant points (Mel scale 🡺 assign importance to some frequencies)  
    Not compare the 1st with the 1st, it could exist some needed translations or insignificant isolated elements.
  + Comparison: « compare most similar words », eventually if we have a great vocabulary, begin with a leak sort.  
    ex : in French, 6/10  
    S-I-S 🡨 delete the similar parts  
    D-I-S  
    ↑compare relevant part
* HMI
  + Graphic  Interface: display syllable (and the spectrogram)
  + « Purchase » Learning : buttons « ok »/ « not ok » and we put the syllable in the database
  + Labyrinth  « game »: we move with our voice  
    / !\ learning the words the gamer will use during the « game »

# Sound recording

Choice of the driver to use, choice of the mike

When we are waiting for a response, we start the sound recording and it keeps going in continues, we remove the irrelevant parts (without speech or noise)

# Sound treatment

## Auto-correlation

## Time split

Time split in slices of 20/30 ms:   
For all time t, we take a signal’s slice which extend from to. So, if we want a signal’s slice of 20ms duration, we take.

We verify the limits, so the signal’s slice we are extracting don’t go out of the borders defined by the signal:

If alors t\_begin=0  
 If alors t\_end=end\_signal

However, the fast Fourier transformation needs a number of points which are a power of 2 in order to perform faster. And as we will use the fast Fourier transformation on the extracted signal’s slices, we completed it (with a n size), by the following of the entry signal, until n is a power of 2.

## Fast Fourier transformation

Fast Fourier transformation for all slices 🡺 put together in a spectrogram (matrix)

# Conclusion