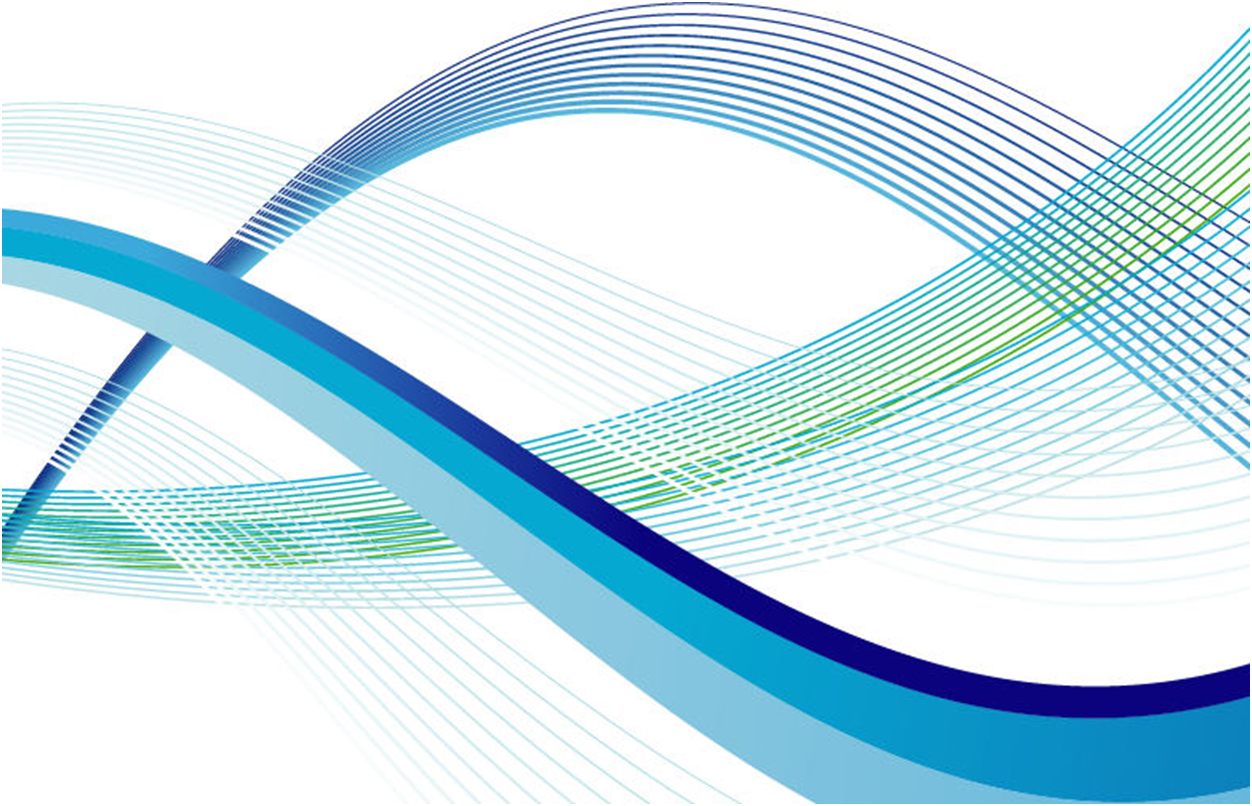
|  |
| --- |
| Polytech’ Nice Sophia Antipolis |
| Word recognition |
| Guénon Marie, Achard Jean-Paul, Favreau Jean-Dominique |
|  |
| **Guénon Marie, Achard Jean-Paul, Favreau Jean-Dominique** |
| **24/1/2013** |

|  |
| --- |
|  |



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 is split in slices of approximately 30 ms:   
For all time t, we take a signal’s slice which extends from to.

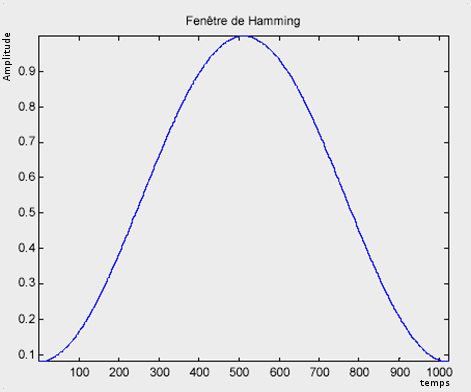
However, the fast Fourier transformation needs a number of points which are a power of 2 in order to perform faster. We chose to take signal’s slices of size n (with n samples), where n is a power of 2 and so that “n\*sampling time” is the closest to 30ms.

We can easily see here that the last signal’s slice will not necessarily has the same size as the others, that’s why we complete the end of the signal with 0 and so have a signal that has a size which is a multiple of the power of 2 considered.

## Spectrogram

### Hamming window

Once the signal is split in slices, we apply to it a Hamming window in order to avoid big discontinuities on the borders, and so avoid inconsistent results with the Fourier transformation.

Formula of the Hamming window:

Where T is the signal’s duration of the segment studied

### Fast Fourier transformation

We are now searching to apply to each signal’s slice the Fourier transformation, which is written:

Where

However, in a goal of speed, we seek to use the fast Fourier transformation, which is written:

For where

For

### Redundancies elimination

Any Fourier transformation is periodic a symmetrical in 0. Since it is not interesting to study the same thing twice (loose of time), we can keep only a signal’s half on which will be made the comparisons.

### Amplitudes weighting

Here we seek to minimize the importance of low frequencies. In fact, the auto-correlation has not removed the whole noise present in the recording and this noise parasitizes the low frequencies.

logarithmic scale to weighting to minimize the   
reduce the dynamic importance of low frequencies

Where

### Mel scale

### Spectrogram reconstruction

The reconstitution of the spectrogram consists of grouping together in a matrix the whole Fourier transformations obtained and transformed for each signal’s instant. The rows of the matrix represent the changes in frequency, the columns the changes in time, and the value of each box represents the amplitude of the Fourier transform on the time and frequency given.

# Conclusion