An isolated word speech to British Sign Language translator

GASNOT Sacha - CAYLA Théo

May 25, 2015

Contents

1 Introduction

The aim of the project was to convert a recorded speech into British-Sign-language pictures, thanks to the analysis of the speech signal. Analyzing a speech signal would make deaf people able to have a direct speech translation on a visual support.

2 Speech signal analysis

2.1 Introduction to speech models

The different phonemes properties have been used to characterize every existing word in the English language. They can be classified in two different sound families : voiced and unvoiced phonemes. A sound is considered as voiced when the vocal chords are used. A voiced phoneme signal shows periodic properties. The analysis of the signal of the pronounced words *"Hello World"* allows to determine the signal of each phoneme present in the sentence. For instance, the sound "H" is unvoiced, and the sound "OW", at the end of the word *"Hello"* is voiced. The microphone is considered perfect, so that the electric noise can be neglected.

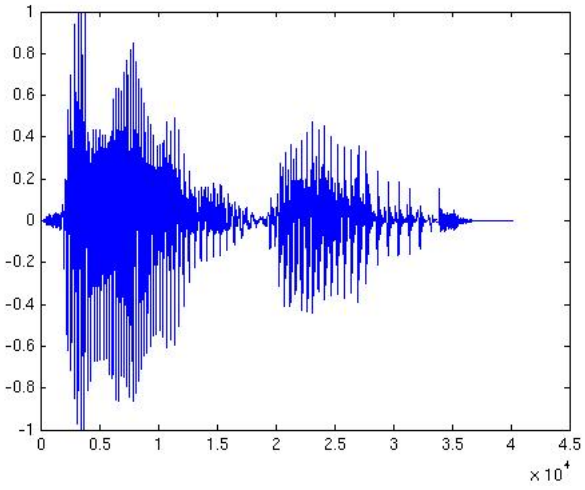

Figure 1: Speech signal containing *"Hello World"*

Speech signal containing *"Hello World"* is given in Figure 1.

The signal is discretised with 44100 samples in order to work with enough discrete values to be accurate, compared to the initial continuous signal.

2.2 Time analysis

Question 1&2 :

The signal x(t) has been sampled during two different periods. These new signals are named x_1 and x_2, and are represented in figure 2 and figure 3. The signal x_1 corresponds to the signal x(t) sampled from t=0,01s to t=0,04s, the signal x_2 has been sampled from t=0,2s to t=0,23s.

x_1 signal shows aperiodic variations, it is an unvoiced signal, whereas a periodic pattern can be seen in x_2; a voiced signal.

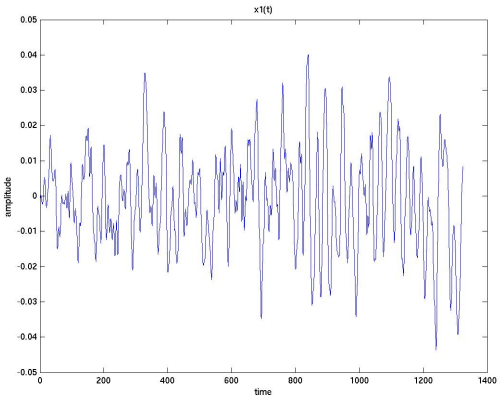

Figure 2: $x_1(t)$

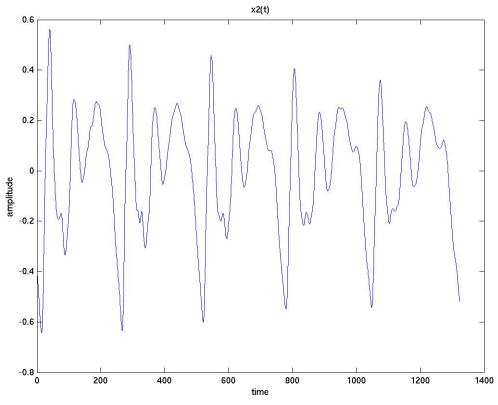

Figure 3: $x_2(t)$

Question 3 :

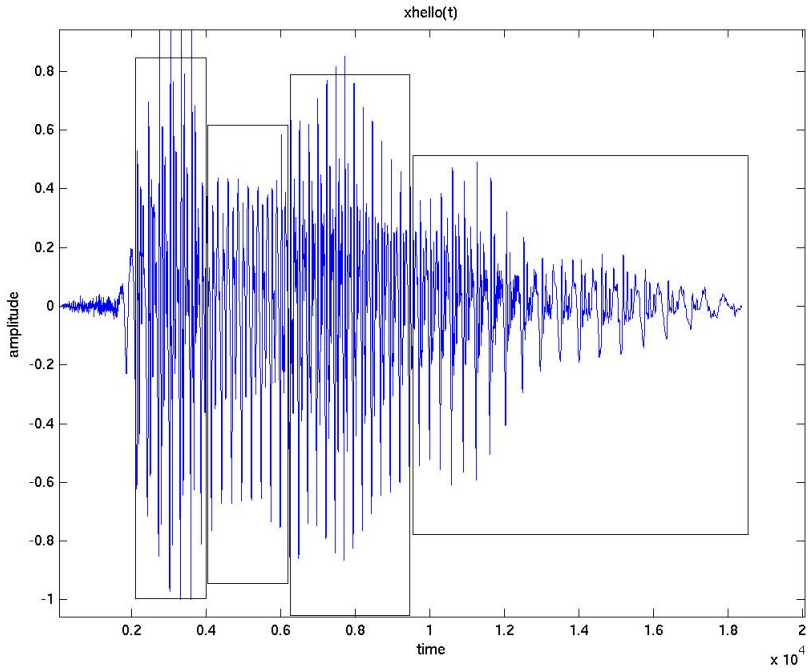

Figure 4: $x_{Hello}(t)$

Four distinct phonemes can be observed in Figure 4, the word is formed by the sounds $[HH, AH, L, OW]$. The signal is shaped differently for each of these phonemes.

2.3 Frequency analysis

Question 4 :

Magnitude spectrum of the samples $x_1(t)$ and $x_2(t)$ have been calculated thanks to their Fourier transform. They are respectively represented in Figure 5. The magnitude spectrum of $x_2(t)$ is thirty times bigger than the magnitude spectrum of $x_1(t)$. The frequencies are contained in $[\frac{-Fs}{2}; \frac{Fs}{2}]$.

2.4 Short Term Fourier Transform and Spectrogram

The study of a Fourier Transform limits the perception of the process, it does not allow the evolution of the spectrum in time to be analyzed. A Fourier transform is represented as a variation depending on frequency. Therefore, the study of the Fourier transform has been held during short periods of time (20 to 30 milliseconds) which have been translated to cover the whole signal, it is a Short Term Fourier Transform.

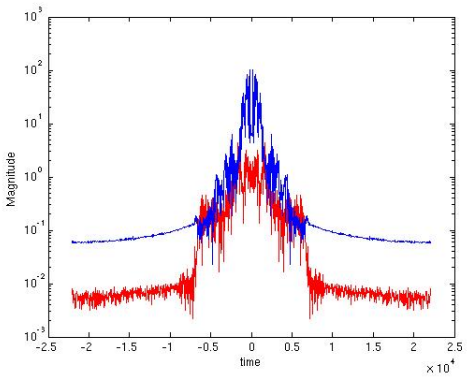

Figure 5: Spectrogram of $X_1(red)$ and $X_2(blue)$

Question 5 :

The L samples of the signal x(t) are considered as a vector of size L. The elements of this vector have been inserted in a matrix of size (N,J), where the first N elements fill the first column, and the N elements following the Mth fill the second (from M+1 to M+N)s, as explained in Figure 6. M is a shift set previously. The empty spots on the last column of the matrix have been replaced by zeros. A column Fourier transform spectrum represents the signal Fourier transform for a range of frequencies, and a line Fourier transform spectrum represents the evolution in time of the entire signal spectrum.

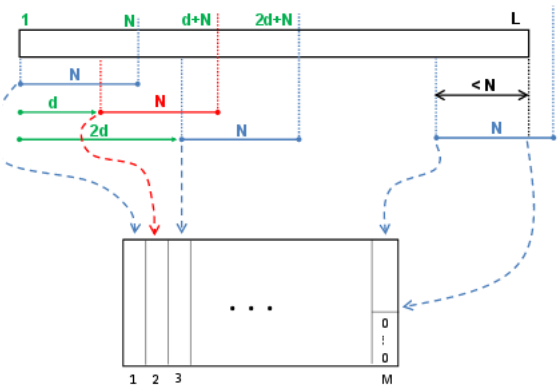

Figure 6: Short Term Fourier Transform

Each row of the matrix then has been multiplied by a Hamming/rectangular window. A Hamming window avoids the signal from being as distorted as it would be when multiplied by a rectangular window.

A column Fourier Transform can be expressed as : $fft(x) \rightarrow [X(0), X(\frac{1}{N}), ..., X(\frac{N-1}{N})]$ The bigger N is, the more accurate the samples are. The discrete Fourier transform (DFT) has been computed on N_fft points so that it would be accurate enough.

Question 6&7 :

A signal spectrogram is calculated from the absolute value of its Fourier transform, squared, divided by the number of samples used in the transform.

$S_x(t,v) = \frac{1}{N}|X(m,v)|^2$

Three spectograms have been plotted, with different parameters, as the number of samples, type of window (Hamming or rectangular).

The values used for the first spectrogram, in Figure 7 are :

$$\begin{cases} N = 441 \\ d = 441 \\ N_{fft} = 1024 \\ w = hamming(N) \end{cases} \tag{1}$$

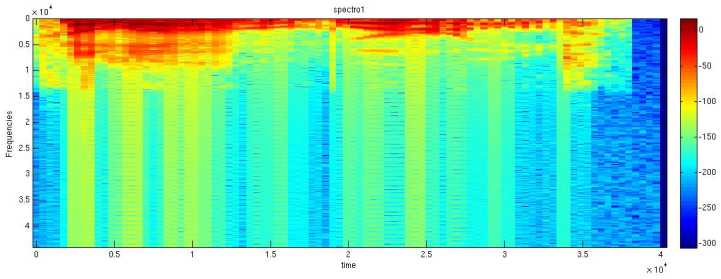

Figure 7: Spectrogram 1

The values used for the second spectrogram, in Figure 8 are :

$$\begin{cases} N = 441 \\ d = 441 \\ N_{fft} = 1024 \\ w = ones(1, N) \end{cases} \tag{2}$$

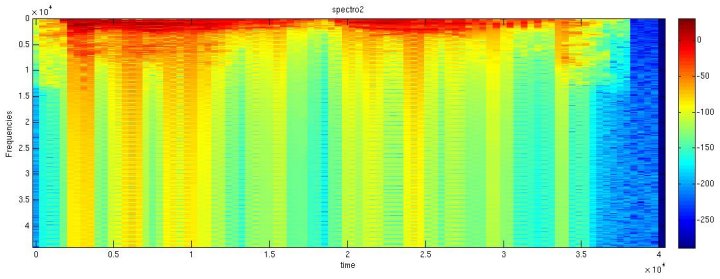

Figure 8: Spectrogram 2

The values used for the third spectrogram, in Figure 9 are :

$$\begin{cases} N = 882 \\ d = 441 \\ N_{fft} = 1024 \\ w = hamming(N) \end{cases} \tag{3}$$

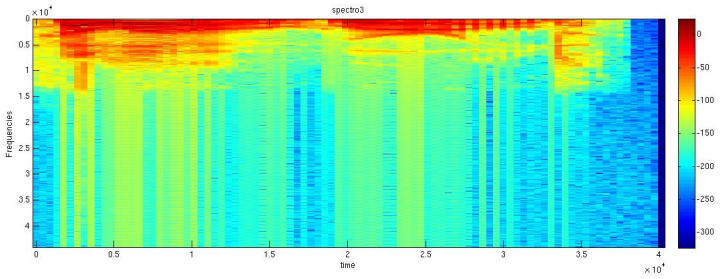

Figure 9: Spectrogram 3

Each spectrogram represents the magnitude of the signal power. The red parts are the most easily heard frequencies. The first and third spectrograms were computed with a similar windowing. The conclusion can be made that a thinner sampling provides a better resolution: the most efficient spectrogram is obtained with a rectangular window because it shows a higher resolution.

3 Feature extraction

3.1 Voiced/voiceless flage and pitch

The aim of this section is to create a function able to differentiate voiced and unvoiced phonemes. When a signal is voiced, the function *isvoiced(x,F_s)* returns 1 and 0 otherwise. A voiced phoneme is defined by the period T of its signal. The pitch of this signal is defined as $p_i = \frac{1}{T}$. An unbiaised estimator is used to determine whether the signal is voiced or not :

$$\gamma_u[p] = \begin{cases} \frac{1}{N-p}\sum_{k=p}^{N-1}(x_i[k]x_i^*[k-p]) & \text{for} \quad \text{p} = \{0,1,..,N-1\} \\ \gamma_u[-p]^* & \text{for} \quad p = \{1-N,...,-1\} \end{cases}$$

A sound is considered voiced if

$\gamma_u[P] >= 0.6\gamma_u[0]$

where P is the position of the maximum.

Then, the pitch of the signal is defined by $p_i = \frac{F_s}{P}$

Question 8&9 :

The function *autocorr(x,maxP)* has been computed, it is similar to the correlation of $x_i[m]$ and itself. The function *isvoiced(x,Fs)* returns 1 if x is voiced, 0 otherwise. autocorr(x,10000) is represented in Figure 10.

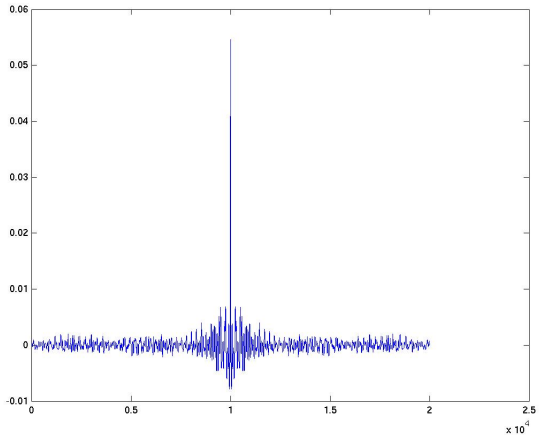

Figure 10: autocorr(x,10000)

The signal x is periodic : it is voiced.

Question 10 :

Two samples of the words *one* and *two* were recorded, the function *isvoiced* determines whether their signals are voiced. The pitch is defined as $p = \frac{1}{T}$

$$\begin{cases} isvoiced(one1, F_s) = 0 \\ isvoiced(one2, F_s) = 0 \\ isvoiced(two1, F_s) = 1 \\ isvoiced(two2, F_s) = 0 \end{cases} \quad (4)$$

The signal *two1* is the only one being voiced; its pitch is equal to 55Hz.
The function *isvoiced* returned different values for two different recordings of the same words. These coefficients are not accurate enough for the application.

3.2 MFCC

Question 11&12&13 :

MFCC features are the result of the signal processing through windows, transforms, and filters. There are five steps in the computing of this function :

MFCC coefficient can't be used because of the different pronunciations' length of the words. An average has to be calculated from the $MFCC_features$ matrix.

- Take the Fourier transform of the signal.
- Compute the spectrum of the signal, filtered with a triangular windows.
- Calculate these signal logarithms
- Take the Discrete Cosine Transform of the log powers.
- The MFCCs are the amplitudes of the resulting spectrum.

$mfcc_i = \sqrt{\frac{2}{P}} \sum_{j=1}^{P} log(E_j) cos(\frac{\pi}{P} i(j - 0,5))$

The function has been computed thanks to the existing function *compute_filter_bank*.
MFCC were computed for each word, the MFCCs corresponding to *"one"* are represented in Figure 11, the MFCCs corresponding to *"two"* are represented in Figure 12. Results can be considered good as the curves of each recording are really close; MFCC coefficients can be used for the application proposed in this work.

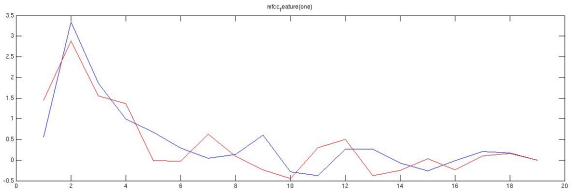

Figure 11: *one*1 and *one*2 signals MFCC coefficients

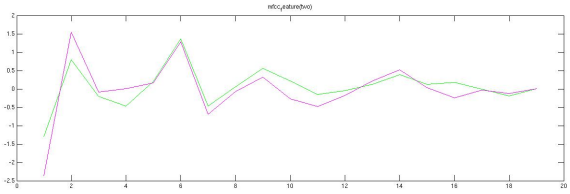

Figure 12: *two*1 and *two*2 signals MFCC coefficients

4 Classification

Classifier train and test

This part consists in recognizing isolated spoken words thanks to the classifier. The *K-Nearest Neighbors (KNN)* algorithm is used, which is divided in two steps :

- The train one : this step consists in extracting the features from the data in the train set.
- The test one : this step creates a distance between features extracted from a signal of the test set and the train set thanks to the Euclidian norm :

  $d(f, f') = ||f - f'||_2 = \sqrt{\sum_k |f_k - f'_k|^2}$

Question 14 :

A *train_ classifier* function is computed to extract the features from the data in the train set. The function returns a matrix containing the features for each element of the train set.

Question 15 :

Then, test_1 and test_2 are used to test the classifier. A class is evaluated thanks to the second step of the KNN algorithm. A probability of success is computed to estimate the correctness of test_1 and test_2 :

$$P(success) = \frac{\text{Number of correctly classified } x \text{ from the test set}}{\text{Total number of elements in the test set}}$$

Probabilities are :

- P1 = 0.58
- P2 = 0.61

Confusion matrix

Question 16 :

The following question consists in computing the confusion matrix, a specific table layout that allows visualization of the performance of a classifier system. It is defined by :

$C_m(i, j) = P(x \text{ is classified in the class } j | x \text{ belongs to the class } i)$

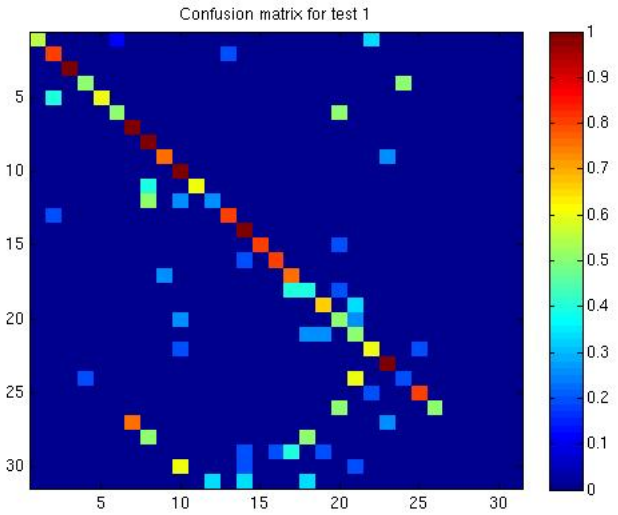

Figure 13: test_1 Confusion matrix

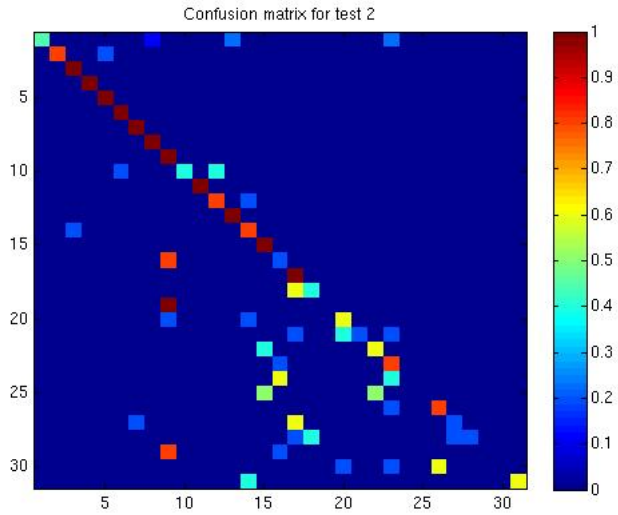

Figure 14: test_2 Confusion matrix

The classes are located on the diagonal of each confusion matrix. Probabilities located on both diagonals are higher than everywhere else on the matrix, which shows that the results are good. Moreover, more classes are found on the confusion matrix of *test_2*, due to more coefficients is *matFtrain2* and *matFtest2*.

The confusion matrix for *test_2* provides better results.

Question 17 :

Each matrix's trace average also makes it possible to compute P(*success*) from the confusion matrix. The results obtained are similar to the results in question 16 :

- P1 = 0.60
- P2 = 0.63