Voice Processing

Marchioro Thomas

Project 0

1 Time-domain processing – VUS discriminator

I sampled the frames using an analysis window of $T_{\text{frame}} = 30$ ms, with a shift of $\tau = 10$ ms between two consecutive frames.
I defined a function `get_metrics`, which computes the energy and the number of zero crossings for each frame.
For an audio signal sampled at frequency $f_s = 16$ kHz, the analysis window contains $L = T_{\text{frame}} f_s = 480$ samples and must be shifted by $U = \tau f_s = 160$ samples between two consecutive frames. If the overall number of samples is D, the number of frames should be D/U. Nonetheless, the last L/U frames cannot be computed, since their length exceeds the maximum index of the sample array, therefore I did not compute them, obtaining a number of frames equal to

$$N_{\text{frames}} = \left\lceil \frac{D - L}{U} \right\rceil . \tag{1}$$

An alternative solution could consist in computing all the D/U frames employing a zero-padding at the end of the file, but this would create some "fake" frames.

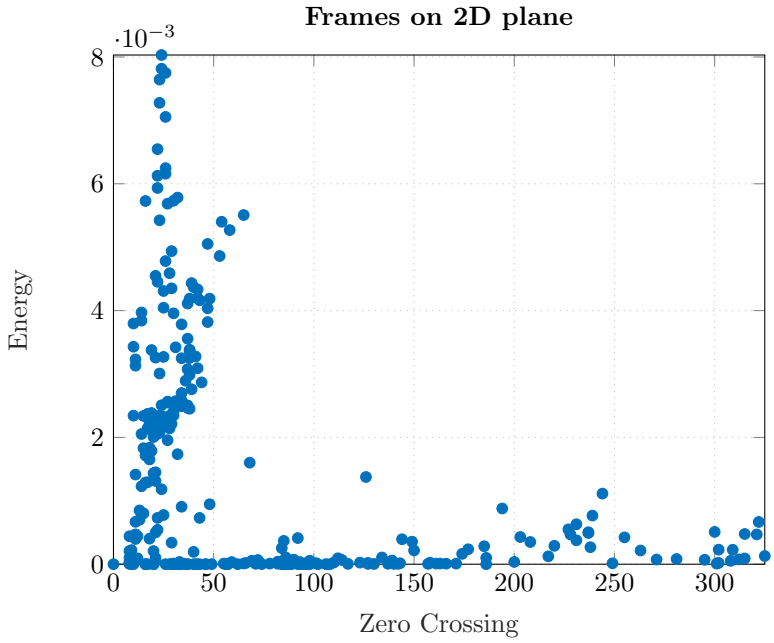

Figure 1: Representation of the frames' energy and zero crossings on a 2D plane.

1.1 Energy and zero crossings

After computing energy and zero crossings for each frames, I plotted them on a 2D plane (see Figure 1). Unfortunately, there was no trivial separation between the data that could be effectively

exploited by basic clustering techniques, thus I opted for a threshold based approach, as suggested in the assignment. The thresholds are based on the following ideas:

- voiced frames have, in general, high energy and low number of zero crossings;
- unvoiced frames have lower energy and high number of zero crossings;
- silence frames have very low energy and low number of zero crossings.

Also, to be applicable to a generic signal, the thresholds must be functions of the mean and standard deviations of the two metrics, which will be denoted as $(\mu^{(\mathcal{E})}, \sigma^{(\mathcal{E})})$ for the energy and $(\mu^{(ZC)}, \sigma^{(ZC)})$ for the zero crossings.
A naive threshold-based implementation consists in defining the following rules:

– Set (V)oiced if $\mathcal{E}_i > \alpha\mu^{(\mathcal{E})} \wedge ZC_i < \beta\mu^{(ZC)} - \gamma\sigma^{(ZC)}$;

– Set (U)nvoiced if $\delta\mu^{(\mathcal{E})} < \mathcal{E}_i < \alpha\mu^{(\mathcal{E})} \wedge ZC_i < \beta\mu^{(ZC)} - \gamma\sigma^{(ZC)}$;

– Set (S)ilence, otherwise.

Nonetheless, I have noticed that there are cases in which some background noise causes a higher number zero crossings: hence, I decided to relax the condition on the zero crossings for voiced signals, and to base their discrimination only on the energy. Then, I checked the values of the energy and zero crossings for some frames that could be easily discriminated and tuned the threshold parameters, obtaining the rules:

– set (V)oiced if $\mathcal{E}_i > 0.6\mu^{(\mathcal{E})}$;

– set (U)nvoiced if $10^{-3}\mu^{(\mathcal{E})} < \mathcal{E}_i < 0.6\mu^{(\mathcal{E})} \wedge ZC_i < 1.2\mu^{(ZC)} - 0.4\sigma^{(ZC)}$;

– set (S)ilence, otherwise.

1.2 Results

Figure 2 shows that these rules lead to a quite reasonable division over the zero crossings – energy plane for the given audio track.

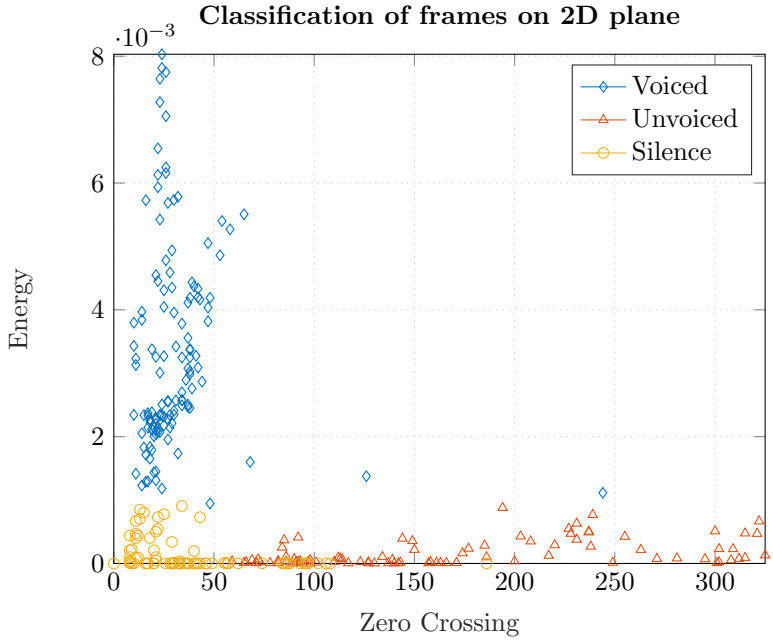

Figure 2: Frames classification on the zero crossing – energy plane.

The final results of the VUS discrimination are reported in Figure 3. Most of the frames are correctly classified.

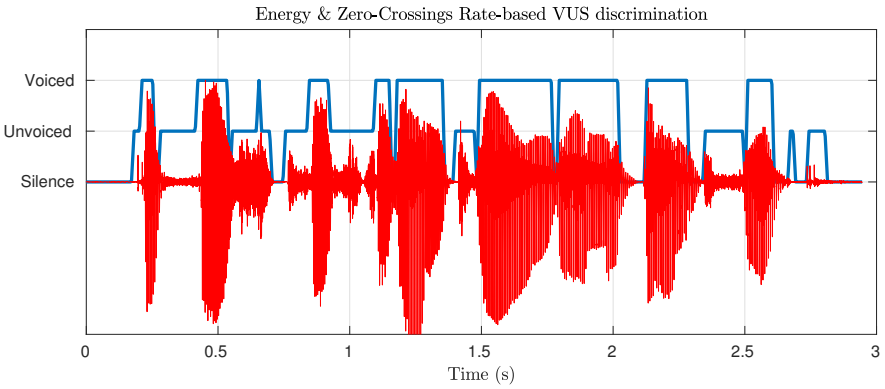

Figure 3: Threshold-based VUS discrimination of the given track.

Changing analysis window and frame rate The choice of 30 ms for the analysis window revealed to be reasonable: I tried to change the frame size to 15 ms and to 60 ms and in both case the results were slightly worse in both cases. With 15 ms, the window was too small, losing information about the context and classifying a silence some lower tones in the unvoiced parts. With 60 ms, instead, the window was too broad and some silent parts were classified as unvoiced or voiced because they were close to tone peaks. Changing the frame rate, instead, determines the number of data points. With a very small frame shift (e.g., 1 ms), a lot of frames are created that are between two classes: a linear interpolation between few frames provides a better description of such points. Nonetheless, too few points lead to inaccurate classification of parts that clearly belong to a specific class.

Recording my own voice I recorded my own voice at 16kHz, while saying "Ok Google, play Despacito", and the VUS discriminator does not perform very well in this case. The main reason is that at the beginning of a sentence my voice tone tends to be lower, hence the "Ok" is classified as unvoiced, being below the energy threshold. Also, in general my voice is not as clear as the reader in the given audio track and the microphone of my device is probably not as qualitative as the one used for recording the other track. On the other hand, both silence and the word "Despacito" are classified quite accurately.

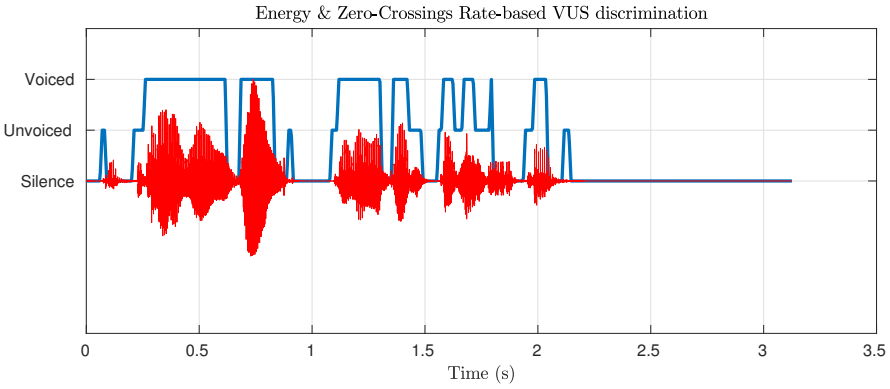

Figure 4: Threshold-based VUS discrimination of my recorded voice.

2 Frequency-domain processing – Age and gender detector

I performed the pitch estimation on the provided waveforms using the autocorrelation method (named ACF in the plots) and the peak picking method on discrete Fourier transform (named FFT in the plots). Before explaining the algorithm I have used for picking the peaks, I need to make some considerations to justify the preprocessing part:

1. the autocorrelation function is always symmetric with respect to $k = 0$, i.e.

$$\phi(k) = \sum_{m=-\infty}^{\infty} x[m]x[m+k] \overset{\ell = m+k}{=} \sum_{\ell=-\infty}^{\infty} x[\ell - k]x[\ell] = \phi(-k),$$

   therefore I analysed only the positive shifts, without loss of generality;

2. the Fourier transform $X(f)$ of a real signal $x[n]$ enjoys hermitian symmetry and, if the discrete signal has sampling frequency F_s, it holds that $|X(F_s - f)| = |X(-f)| = |X(f)|$, therefore the frequencies above $F_s/2$ can be ignored in the analysis of the FFT.

Peak picking algorithm The peak picking algorithm that I have used is based on a simple idea: the first peak of a function is the first point x_i above a certain threshold satisfying the local maximum condition:

$$f(x_i) > f(x_{i-1}) \wedge f(x_i) > f(x_{i+1}), \tag{2}$$

i.e., the first point that has a higher value of f than its two neighbours. I decided to employ a threshold to avoid detecting small lobes which are not actual peaks. I found that reasonable threshold values are 0.98 for the autocorrelation method and 0.1 for the fft method. Some examples of peaks found on a single frame using the two methods are shown in Figures 5 and 6. In particular, for a frame sampled from `ae-pout-255Hz-8kHz.wav`, the estimated peaks were 258.1 Hz with the autocorrelation method and 260.7 Hz with the fft method, which are reasonably close to the correct value of the pitch.

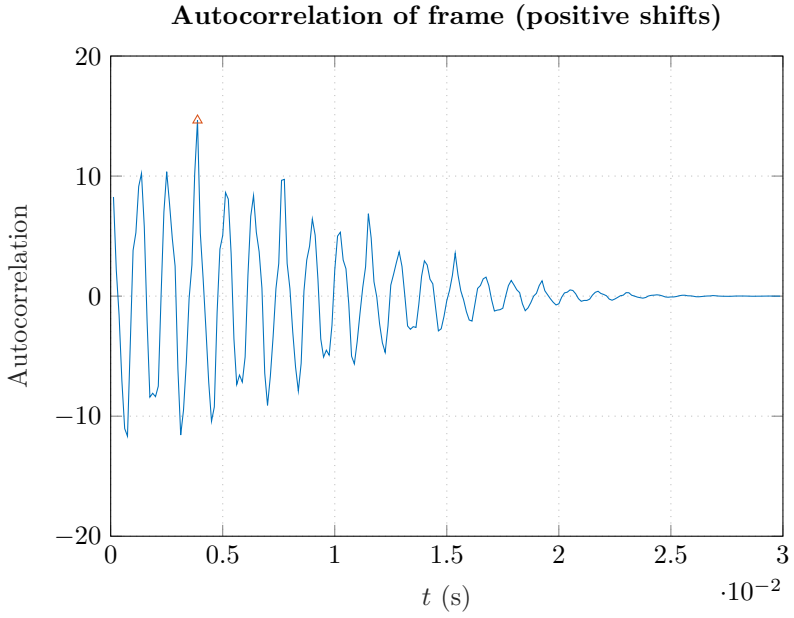

Figure 5: Example of peak found using the autocorrelation method.

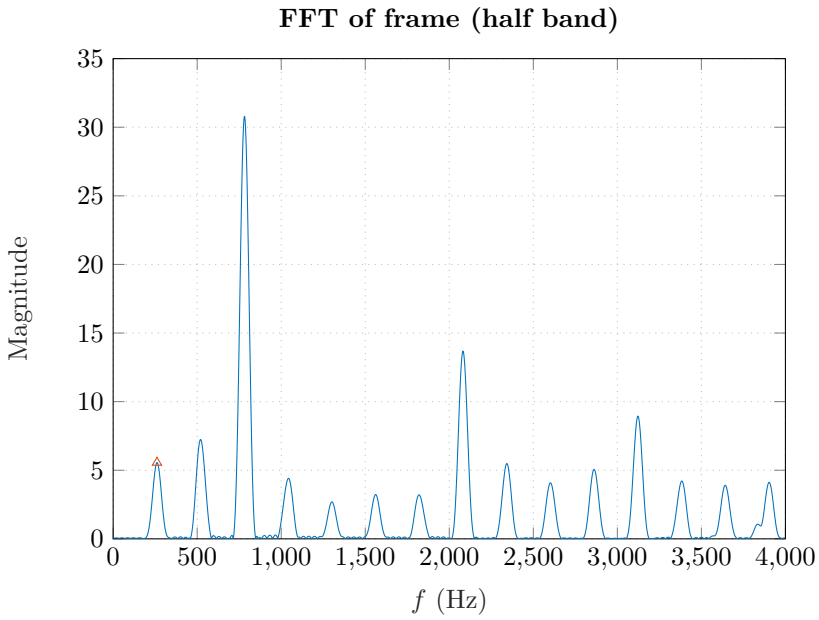

Figure 6: Example of peak found using the fft method.

Age and gender detection Hence, I implemented some discrimination rules to determine the age and gender of a person, based on the estimated pitches. The core idea is to consider only the estimated peaks in the range $70 - -500$, computing the average μ_{pitch} and doing the classification as follows:

- set "adult male" if $\mu_{\text{pitch}} < 155$ Hz;
- set "adult female" if $155 \text{ Hz} < \mu_{\text{pitch}} < 275$ Hz;
- set "child" if $\mu_{\text{pitch}} > 275$ Hz.

2.1 Results

I applied the two methods to estimate the pitch on each frame of `ae-pout-255Hz-8kHz.wav` and interpolated the results using splines. The result was close to a straight line, as expected since the audio consists in a single long vowel. This is confirmed also by comparing the pitch plot to the spectrogram (Figure 7). Worse results are obtained by applying the methods to the audio file `H.22.16k.wav` (Figure 8), in particular with the autocorrelation method. Nonetheless, the fft method is able to correctly classify the age and gender of the speaker.

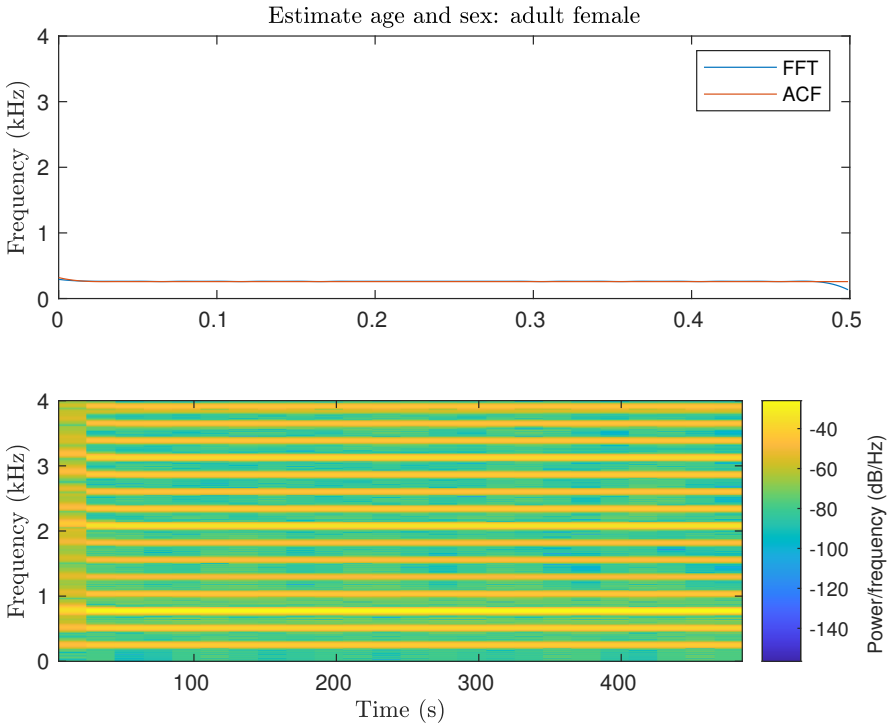

Figure 7: Threshold-based VUS discrimination of my recorded voice.

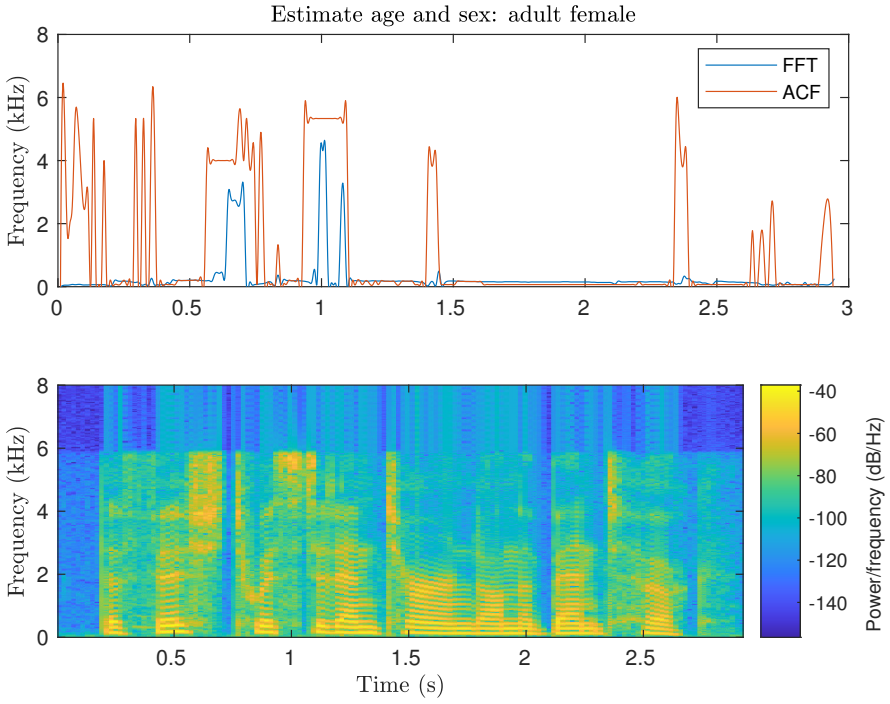

Figure 8: Threshold-based VUS discrimination of my recorded voice.