Annales UMCS Informatica AI 8(1) (2008) 157-163
10.2478/v10065-008-0015-3

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Utterance intonation imaging using the cepstral analysis

Ireneusz Codello*, Wiesława Kuniszyk-Jóźkowiak,
Tomasz Gryglewicz, Waldemar Suszyński

Institute of Computer Science, Maria Curie-Sklodowska University,
pl. M.Curie-Skłodowskiej 1, 20-031 Lublin, Poland

Abstract

Speech intonation consists mainly of fundamental frequency, i.e. the frequency of vocal cord vibrations. Finding those frequency changes can be very useful – for instance, studying foreign languages where speech intonation is an inseparable part of a language (like grammar or vocabulary). In our work we present the cepstral algorithm for F0 finding as well as an application for facilitating utterance intonation learning.

1. Introduction

We can divide human speech into two categories:

– voiced speech – the air from lungs causes vocal cords vibration. The frequency of these vibrations is called fundamental frequency, vocal tone or zero formant (F0);
– unvoiced speech – the air from lungs goes untouched throughout vocal cords. No vibrations are caused, therefore no fundamental frequency is created.

As we can see in Fig. 1, the vowel ‘a’ as an example of voiced speech, is very regular (due to regular vocal fold vibrations) contrary to the consonant ‘s’, which is very irregular, noisy (due to noise excitation – by the air from lungs untouched by vocal folds).

Fundamental frequency determines the intonation of speech. These intonation changes (increasing, decreasing) can have huge influence on the meaning of a spoken sentence – for example, we can distinguish a question from an ordinary sentence. We can recognize intentions of a speaker, whether he is mad, polite or curious. In many languages (English, Japanese) intonation (like vocabulary or grammar) is an inseparable part of language.

*Corresponding author: *e-mail address*: irek.codello@gmail.com

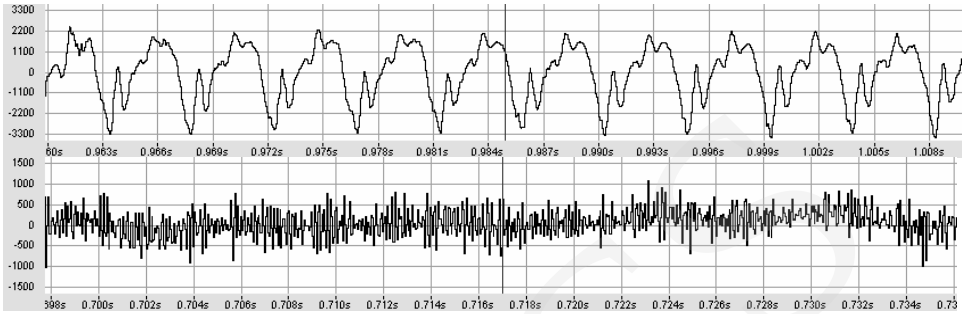

Fig. 1. Oscillogram of the vowel 'a' (top) and the consonant 's' (bottom)

2. Computation procedure

Human vocal tone varies between 50 Hz and 1000 Hz:

- 50 Hz – 250 Hz – ordinary speaking man,
- 150 Hz – 350 Hz – ordinary speaking women,
- 300 Hz – 500 Hz – ordinary speaking child,
- up to 1000 Hz – opera singer (soprano).

The cesptral analysis needs a few periods of vocal cord vibrations to determine it in speech. The signal of 50 Hz – 500 Hz frequency has a period between 20 ms and 0.5 ms, therefore the cepstral analysis computation frame has to last from 40 ms even to 100 ms (if we expect to analyze mele voice).

The basic cepstral analysis algorithm consists of the following steps:

1) windowing – we divide the signal *x(t)* into frames (windows) of the same length. Consecutive frames can overlap each other (usually with 50% frame length). After that each frame is analyzed independently of the other ones. Then the frame is multiplied by the window function (for instance Hamming window);
2) FFT – we compute frame spectrum using Fast Fourier Transform;
3) filtering – we can filter the spectrum *X(t)* (in our work we use a low-pass filter with 5,5kHz cut-off);
4) decibels – we change the amplitude scale of $X(t)$ from linear to logarithmic. Because we use a real cepstrum (instead of a complex one) we compute a real logarithm using the equation:

$$\begin{aligned} Y(k).re &= 20\log_{10}\left(\sqrt{X(k).re^2 + X(k).im^2}\right) \\ Y(k).im &= 0 \end{aligned} \tag{1}$$

where $X(k)$ – k-th complex spectral line of the frame

instead of the complex logarithm:

$$Y(k).re = 20\log_{10}\left(\sqrt{X(k).re^2 + X(k).im^2}\right)$$
$$Y(k).im = arctg\left(\frac{X(k).im}{X(k).re}\right) \tag{2}$$

5) iFFT – we compute an inverse FFT of $Y(k)$ obtaining the frame cepstrum $C(k)$;
6) F0 finding – we find an extremum of the cepstrum within a range of 50 Hz – 1000 Hz.

The cepstrum horizontal axis is time t which can be easily transformed into herz f using the formula:

$$f = \frac{1}{t} \tag{3}$$

The final result is a graph of F0 changes, where on the X-axis we put time (consecutive frames) and on the Y-axis we put frequency (extremum frequency of each frame cepstrum).

3. An example

Here we have an exemplary utterance – the vowel 'a' said by a female (her voice intonation is increasing and then decreasing in time).

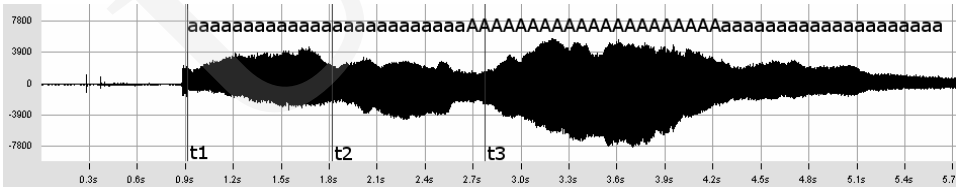

Fig. 2. The source signal – the Polish vowel 'a' said by a female. Her voice intonation is increasing and then decreasing in time

Let us choose three arbitrary frames t_1, t_2, t_3 (vertical lines in Fig. 2) and then compute its' Fourier Transform (46ms frame length, 25% window length overlap, Hamming window).

We can see regular of amplitude fluctuations and a period of those fluctuations. This period is the fundamental frequency and can be obtained from the inverse of FFT of those frames. In Fig. 4 we can see those cepstrums – each with the maximum amplitude and its time transformed into frequency (equation (3)). We can also see that the beginnings and ends of those graphs are equal to zero. These sections were set to zero due to the fact that in the (0.1)ms range corresponding to (+∞,1000)Hz and in the (20.23)ms range corresponding to (50.43)Hz there is no base tone.

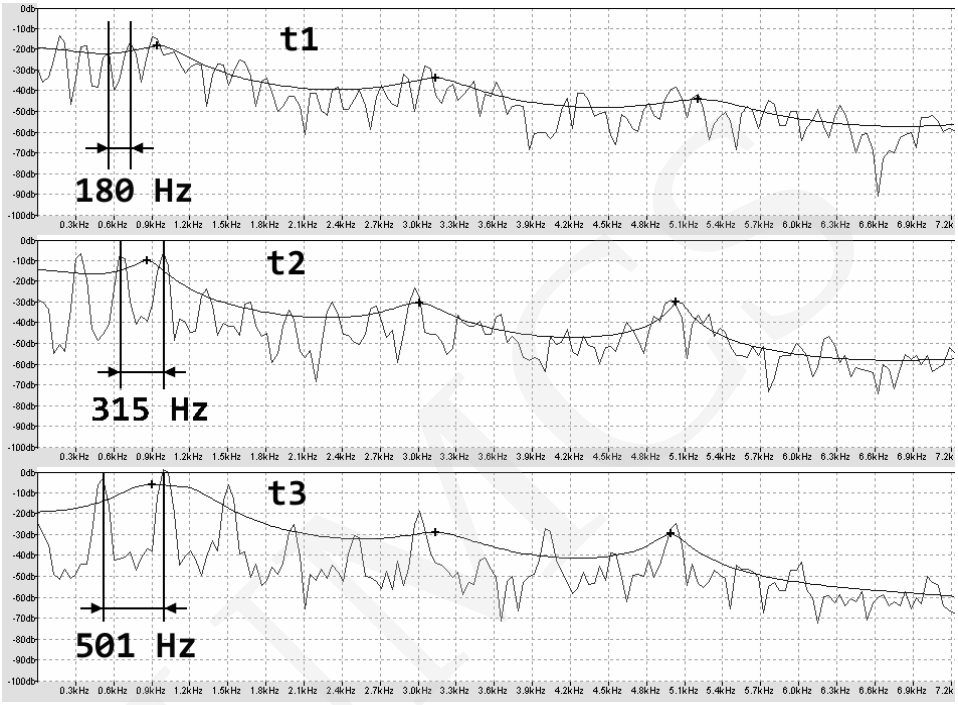

Fig.3. Spectra of the frames t1, t2, t3 of the source signal

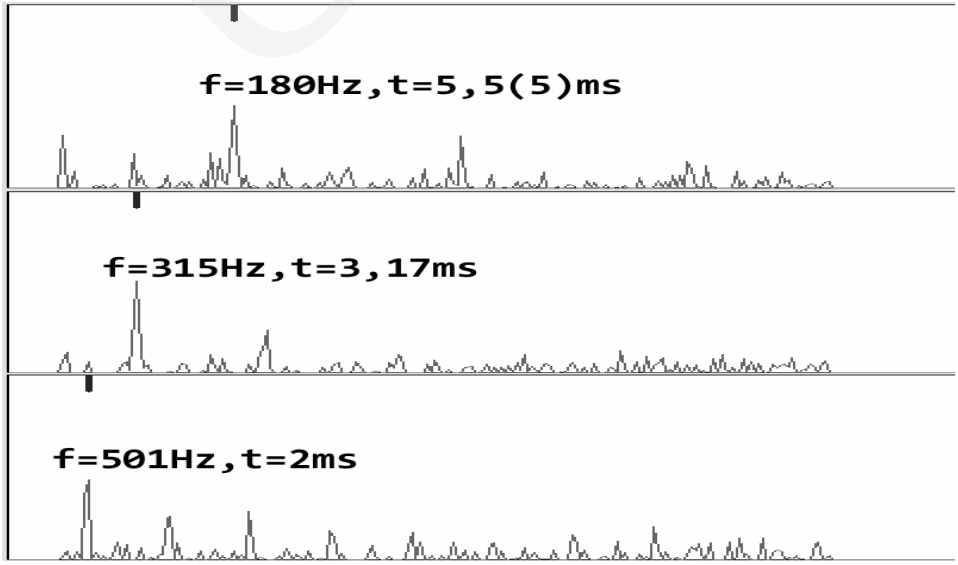

Fig. 4. Cepstrums of the frames t1, t2, t3 of the source signal. The X-axis is the time from the range (0.23)ms. (23.46)ms range is a mirror reflection due to the Fourier property and thus it is not depicted in the graph

By combining all extrema (one from every cepstrum) we obtain our result – F0 changes the graph.

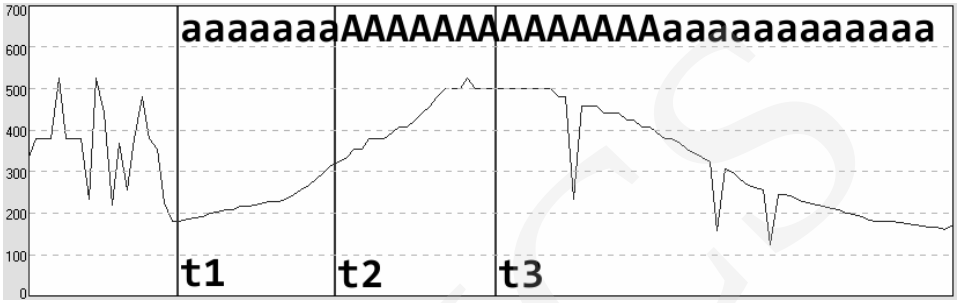

Fig. 5. F0 changes in time of the source signal

As we can see in Fig. 5, the result is not clear. Firstly, we see the F0 before t_1 frame, where there is no signal – so the silence detection must be made. In our work we simply compute envelope of the signal and assume that the silence envelope is less than some value (threshold) which is the input parameter of an algorithm. Secondly, not all cepstrums have their extrema corresponding to the base tone – that is why there is some discontinuity after t_3 frame. Therefore we need to use some sort of filtering to smooth the result – for instance we can use a low-pass filter.

There is the third problem. The input signal can contain not only silence parts and voiced speech but also unvoiced speech as well as noisy speech. Unvoiced speech has no base tone, therefore it has to be treated as a silence – the cepstrum maximum should not be taken into account. Noisy speech is problematic too – it has additional frequencies which can be taken as base tone. Besides envelope, there are a few more factors that can be useful in F0 filtering, like:

- signal oscillation number per frame – we can roughly distinguish voiced and unvoiced speech,
- SNR of a cepstrum – we can estimate the quality (significance) of the cepstrum maximum (whether it is above other peaks the cepstrum or not),
- a number of high local extrema in the cepstrum – we can count a number of significant extrema in the cepstrum (when there are many extrema there is greater probability that the highest one is not a base tone),

but research on their usability is still in progress.

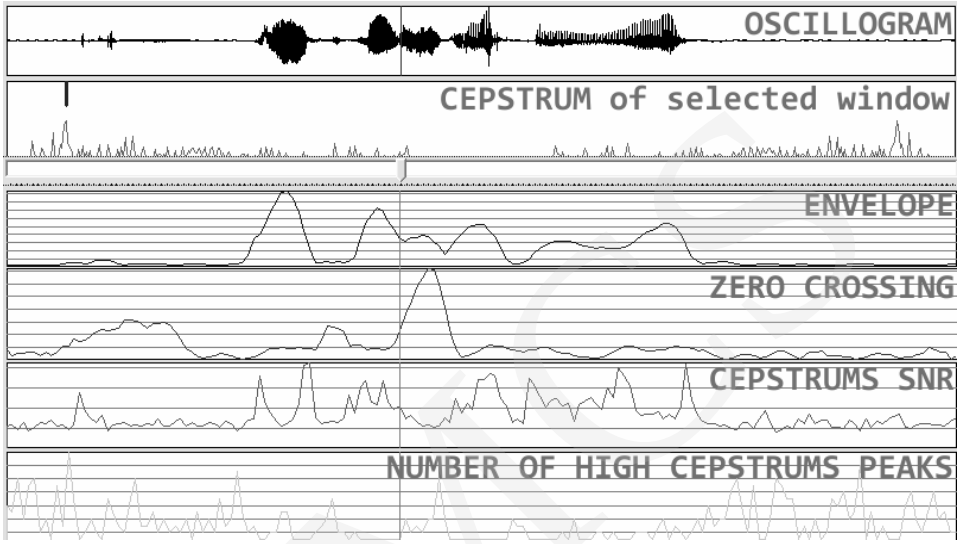

Fig. 6. Potentially useful coefficients for F0 tracking

4. Application

We developed a simple tool for speech intonation learning.

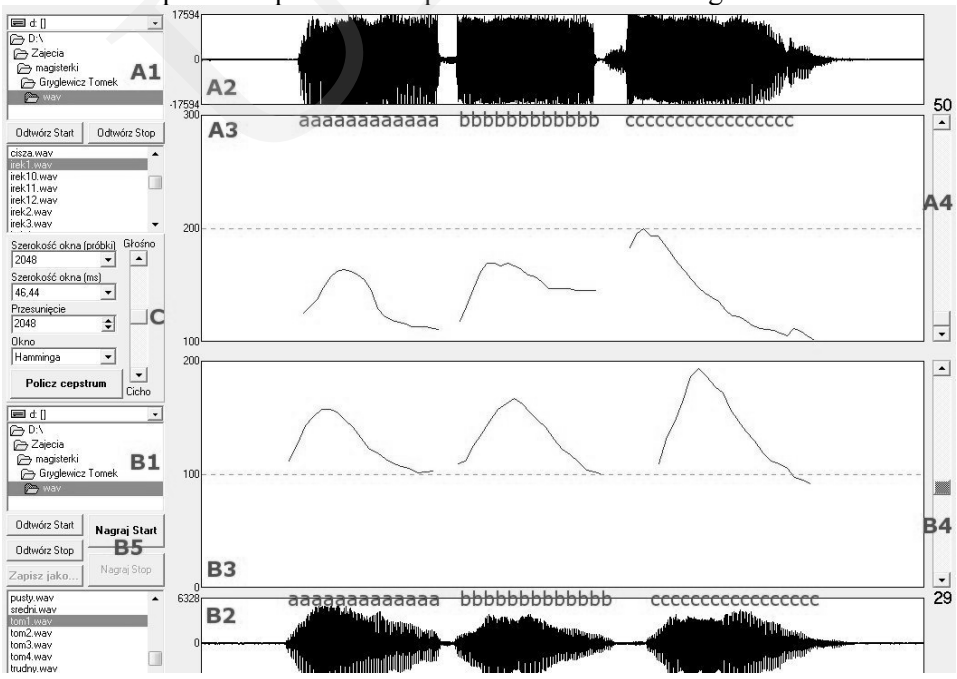

Fig. 7. The application screenshot. The parameters of an algorithm: Hamming window, frame length – 46 ms, overlap – 100%. The input signal: „aaaaa bbbbb ccccc" said by two men

It is divided into 3 sections: teacher – A, student – B and algorithm – C. A user can open teacher'y speech file in section A and his own speech file in section B. Then he can change algorithm parameters in section C like a window width (in samples or in milliseconds), frame overlap, window function and volume of speech playing. While computing the cepstrum (button in section C) a user can compare teacher's intonation A3 with his own B3. Moreover, he can change envelope threshold of both files independently by A4 and B4 scrolls making the graph clearer (as discussed in section 3 in this article). Of course, one can record the speech samples by B5 buttons which later can take part in intonation comparison.

From the example in Fig. 7 we can see that utterance intonations "aaaaa bbbbb ccccc" of the teacher and the student roughly match each other. It is the sign for the student that he said it correctly and could pass to the next sample.

Conclusions

In our opinion the software is very helpful in intonation learning. Intonation comparison is easier and more objective if it is based on intonation graph rather than on hearing. As a consequence, one can learn alone (without a teacher) more often – undoubtedly, it is a big advantage.

References

[1] Rabiner L.R., Schafer R.W. *Digital Processing of Speech Signals*. New Jersey, Prentice-Hall, Inc., (1978).
[2] Gold B., Morgan N., *Speech and Audio Signal Processing*. John Wiley & Sons Inc., New York, (2000).
[3] Basztura Cz., *Źródła, sygnały i obrazy akustyczne*. Wydaw. Komunikacji i Łączności, Warszawa, (1988), in Polish.
[4] Tadeusiewicz R., *Sygnał mowy*. Wydaw. Komunikacji i Łączności, Warszawa, (1988), in Polish.
[5] Pawłowski Z., *Foniatryczna diagnostyka*. Oficyna Wydawnicza Impuls, Kraków, (2005), in Polish.