Speech sample salience analysis for speech cycle detection

C. Mertens.1, F. Grenez.1, J. Schoentgen.1,2

1Laboratory of Images, Signals and Telecommunication Devices, CP 165/51, Faculté des Sciences Appliquées, Université Libre de Bruxelles, 50, Av. F.-D. Roosevelt, B-1050 Brussels, Belgium
2National Fund for Scientific Research, Belgium

chmerten@ulb.ac.be, fgrenez@ulb.ac.be, jschoent@ulb.ac.be

Abstract

The presentation proposes a method for the measurement of cycle lengths in voiced speech. The background is the study of acoustic cues of slow (vocal tremor) and fast (vocal jitter) perturbations of the vocal frequency. Here, these acoustic cues are obtained by means of a temporal method that detects speech cycles via the so-called salience of the speech signal samples. The method does not request that the signal is locally periodic and the average period length is known a priori. Several implementations are considered and discussed. Salience analysis is compared with the auto-correlation method for cycle detection implemented in Praat.

Index Terms: vocal frequency, vocal jitter, speech sample salience analysis

1. Introduction

In clinical applications of speech analysis, speech cycles are detected to measure their lengths and amplitudes with a view to investigating slow (vocal tremor) and fast (vocal jitter and shimmer) perturbations of vocal frequency and speech cycle amplitude. Often, such analyses are frame-based and the cycle detection rests on relevant signal peaks that have to be selected among several candidates.

To facilitate this selection, one often assumes that the peaks are regularly spaced in time so that they can be determined one by one on the basis of a prior estimation of the typical fundamental period. The assumption of quasi-equal spacing is, however, valid for modal voices only and not for pathological ones. Cycle insertion or omission errors may therefore occur, which bias the acoustic cues of cycle regularity.

Here, we propose to track speech cycles via the so-called sample salience. Salience analysis does not rest on the assumptions that the signal is locally periodic and the average period length is known a priori.

2. Salience analysis

2.1. Definition

One considers a speech signal array of length M. The salience s(k) of sample k is defined as the length of the longest interval over which that sample is a maximum. One defines also right salience s_r(k) and left salience s_l(k) as the number of samples over which sample k is a maximum to the right and left. Total salience and left and right saliences are related as follows.

\[ s(k) = s_l(k) + s_r(k) + 1 \]  \hspace{1cm} (1)

Therefore, the salience of the global maximum is M, if there is no other sample with the same amplitude. But, a sample with a large salience has not necessarily a large amplitude and vice versa. For instance, Figure 1 shows that the 4th sample, in spite of the fact that it has a larger amplitude than the 7th, has a smaller salience.

![Figure 1: Example of signal sample saliences](image)

2.2. Basic algorithm

The goal is to assign a salience value to each sample. The estimation of the salience consists in considering all possible within-array analysis intervals and recording the length of the largest interval over which a sample is a maximum. The calculation of the sample salience therefore involves the following steps.

1. Initialization : all sample saliences are put equal to 1 (because each sample is a local maximum with regard to itself) and the length of the analysis interval is put equal to n = 2.
2. Subdivision of signal array of length M into analysis intervals of length n. The rightmost interval stops at the right array boundary whatever its length (i.e. the rightmost interval length is comprised between 1 and n).
3. Determination of the maximum within each interval and assignment of a salience of n to the interval maximum.
4. Increase of the interval length n by one and looping back to step 2.

At the end, all the samples have at least one and maximally M salience values. Only the highest salience...
value is kept for each sample. As an example, Figure 2(a) outlines the different steps of the algorithm for an array of length 16.

However, the speech sample salience values thus obtained may be biased. The reason is the arbitrary position of the array with regard to the analyzed signal. This is a known problem in multi-scale signal analysis, of which salience analysis is an example. For instance, in Figure 2(a), the 2nd sample has a salience equal to 1. However, if the second sample of the array were taken as the origin, it would have a salience equal to 4. A related problem is that no information is available about the samples outside the analysis array. As a consequence, spurious salience values may be assigned to the samples near the boundary of the array.

2.3. Within-array rotation

Boundary effects may be taken into account by rotating $M$ times the samples of the array, so that each sample occupies once the left and right boundary positions. The final sample saliences are the average saliences, which may be considered to be independent of their positions with regard to the array boundaries [1]. However, when rotating, samples that are distant in time may be put into contact, which may be a different cause of bias of the sample saliences. Rotation also increases processing time. For these reasons, an alternative, described below, has been implemented and tested.

2.4. Partial salience allocation

To speed up processing, the partial salience allocation analysis exploits a feature of the tabular representation illustrated in Figure 2(a). Instead of computing maxima over intervals of increasing length 2, 3, ..., one may determine the maximum over frames of decreasing lengths.

Let $j$ be the position of the global maximum of the array. One knows that this sample is a maximum over the $j - 1$ previous samples so that, at the next step, one may compute the maximum over the $j - 1$ first samples only, and assign it a salience value of $j - 1$ and so on, until a salience has been assigned to the first sample (which is guaranteed to receive a salience value via that procedure). For each sample to which a salience is thus attributed, one also stores the left and right saliences.

Figure 2(b) illustrates the application of that rule to an array, which is the same as in Figure 2(a).

2.5. Sliding analysis window

However, all samples are not assigned a salience value via that scheme and the problems related to the array origin are not solved either. Therefore, we have used a sliding window $w_N(i)$ (of length $N < M$ and origin $i$), which is placed at the beginning of the array and moved sample-by-sample to the end. Saliences $s(k, i)$, $s_l(k, i)$ and $s_r(k, i)$ are then obtained with reference to the sliding window. If the analysis window length is well-chosen, the running saliences so acquired are adequate for most purposes.

Hereafter, we distinguish local saliences defined with regard to one given window position and running saliences $s^*(k)$, $s^*_l(k)$ and $s^*_r(k)$ defined as the maximum of the local saliences of sample $k$ for all the previous and present window positions. Therefore, running salience values can only increase, but not decrease. The steps involved are the following.

1. Initialisation: all saliences $s^*$ are put equal to 1 and all saliences $s^*_l$ and $s^*_r$ equal to zero.
2. Application of the partial salience allocation for position $i$ of the sliding window, obtaining the local

Figure 2: Examples
3. Application to speech cycle length
   extraction in voiced speech sounds

3.1. Preprocessing

To speed up salience analysis, the speech signal has been decimated to a sampling frequency of $F_s = 8000\text{Hz}$. Assuming that the vocal frequency is comprised between $F_{0\min} = 60\text{Hz}$ and $F_{0\max} = 400\text{Hz}$, the length $N$ of the sliding analysis window has been chosen to be 50% larger than the longest possible cycle.

$$N = 1.5 \cdot \frac{F_s}{F_{0\min}} \tag{3}$$

Before salience analysis, the signal has been band-pass filtered by means of a finite impulse response (FIR) filter with cutoff frequencies equal to $60\text{Hz}$ and $1000\text{Hz}$ to remove low-frequency hum, additive noise owing to turbulence as well as high-frequency formants. Also a first-order integrator has been used to suppress the radiation at the lips. Hereafter, vocal cycle detection is based on the prominent peaks of the speech cycles. Therefore, only the saliences of the signal peaks have been kept for further processing. A signal peak is a sample the left and right neighbours of which have a lower amplitude.

3.2. Detection of speech cycle peaks

One assumes that the prominent speech cycle peaks are characterized by large salience values. The sequence of relevant peaks has been chosen among all possible peaks by means of the coefficient of variation of the inter-peak durations. The coefficient of variation is the quotient of the standard deviation of the inter-peak durations and their mean. The steps are the following.

1. Discarding of peaks that have a salience below twice the length of the shortest possible cycle. Sorting of the remaining peaks in descending order of salience.

2. Selection of an initial number of peaks equal to the speech signal length divided by the largest expected cycle length (i.e. 16ms) minus 1.

3. Insertion of one additional peak into the sequence in order of decreasing salience.

4. For each peak sequence, test of all sequential inter-peak durations whether they are larger than the shortest possible cycle length (i.e. 2.5ms). If not, the intruding peak is removed. The intruding peak is detected by computing the absolute value of the second-order differences between left and right adjacent peak positions. The peak for which this difference is largest is discarded. This rule guarantees the removal of all isolated spurious peaks (i.e. those that do not appear in pairs, triplets, etc).
Table 1: Voice report: vocal frequency and vocal frequency perturbations

<table>
<thead>
<tr>
<th>Signal</th>
<th>Mean $F_0$ [Hz]</th>
<th>Std deviation [Hz]</th>
<th>Jitter (ddp) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>salience analysis</td>
<td>Praat</td>
<td>salience analysis</td>
</tr>
<tr>
<td>signal 1</td>
<td>99.86</td>
<td>99.84</td>
<td>0.60</td>
</tr>
<tr>
<td>signal 2</td>
<td>99.94</td>
<td>99.91</td>
<td>0.54</td>
</tr>
<tr>
<td>signal 3</td>
<td>99.72</td>
<td>99.76</td>
<td>0.98</td>
</tr>
<tr>
<td>signal 4</td>
<td>100.24</td>
<td>100.20</td>
<td>1.32</td>
</tr>
<tr>
<td>signal 5</td>
<td>100.16</td>
<td>100.14</td>
<td>1.54</td>
</tr>
<tr>
<td>signal 6</td>
<td>99.56</td>
<td>99.62</td>
<td>1.93</td>
</tr>
<tr>
<td>signal 7</td>
<td>99.86</td>
<td>99.70</td>
<td>2.22</td>
</tr>
</tbody>
</table>

5. Computation of the coefficient of variation of the inter-peak durations of the peak sequence.
6. Reinsertion of intruding peaks removed sub 4 and looping to step 3. The algorithm stops when all peaks of sufficient salience have been processed.

The peak sequence giving rise to the minimal coefficient of variation is retained. Salience analysis is performed twice, once for each polarity of the signal (peaks and valleys) and the polarity giving the smallest coefficient of variation is kept.

3.3. Cycle duration
The speech signal is upsampled subsequently to $F_s = 200kHz$ to guarantee a high enough precision of the peak positions, given the size of vocal jitter expected in modal voices. The earlier peak positions are refined by searching for the local maxima of the upsampled signal in the vicinity of non-upsampled peak positions. The cycle durations are obtained via the upsampled inter-peak durations. As an example, Figure 3 shows the saliences of the signal peaks for a fragment of a male vowel [a] and the cycle-by-cycle vocal frequency obtained by minimizing the coefficient of variation of the inter-peak durations.

3.4. Jitter
Cycle length jitter designates small random perturbations of the speech cycle lengths. The jitter formula used here reports the second-order difference of the cycle durations:

$$\text{jitter} [\%] = 100 \cdot \frac{\sum_{i=2}^{K-1} |2T_i - T_{i-1} - T_{i+1}|}{\sum_{i=2}^{K-1} T_i}$$

where $T_i$ is the duration of the $i^{th}$ cycle and $K$ is the number of speech cycles in the sequence.

3.5. Corpus
The corpus has comprised seven sustained vowels [a], synthesized by means of a simulator of disordered voices [2]. The signals have been generated with increasing amounts of jitter, which in the framework of the synthesizer consists in perturbing by white noise the instantaneous frequencies of the harmonic driving functions of the synthesizer’s glottal area function model. The synthetic jitter thus defined is a stochastic variable the value of which must be measured a posteriori.

4. Results
Jitter analyses based on salience have been compared to jitter analyses implemented in Praat. The method used in Praat is a short-term analysis resting on the autocorrelation function [3]. Table 1 reports the average vocal frequency, the standard deviation of the cycle durations and the jitter (in percent) obtained via salience analysis and Praat. One observes in Table 1 that the values of jitter in percent obtained via both methods evolve together, but are not identical.

5. Discussion and conclusion
The differences between jitter values (re two rightmost columns in Table 1) are a consequence of the cycle length detection, which is not identical for both methods. Inspecting the cycle markers in Praat, for instance, shows that Praat does not exactly detect the most salient cycle peaks. The advantage of salience analysis is the lack of assumptions with regard to cycle regularity or a priori knowledge of the typical cycle length. The only parameter which depends on vocal frequency is the length of the sliding analysis window, which is fixed once and for all.

6. References