SPEECH RATE DETERMINATION BY VOWEL DETECTION ON THE MODULATED ENERGY ENVELOPE

Tomas Dekens[1,2]*, Heidi Martens*[3]*, Gwen Van Nuffelen*[3]*, Marc De Bodt*[3,4] *and Werner Verhelst*[1,2]

[1]Vrije Universiteit Brussel, dept. ETRO, Pleinlaan 2, B-1050 Brussels, Belgium
[2]iMinds, Dept. FMI, Gaston Crommenlaan 8, B-9050 Gent-Ledeberg, Belgium
[3]Antwerp University Hospital, Rehabilitation Centre for Communication Disorders, Wilrijkstraat 10, B-2650 Edegem, Belgium
[4]Ghent University, dept. of Speech and Language Pathology and Audiology, De Pintelaan 185, B-9000 Ghent, Belgium

ABSTRACT

In this paper we propose a new algorithm to detect vowels in a speech utterance and infer the rate at which speech was produced. To achieve this we determine a smooth trajectory that corresponds to a high frequency energy envelope, modulated by the low frequency energy content. Peak picking performed on this trajectory gives an estimate of the number of vowels in the utterance. To dispose of falsely detected vowels, a peak pruning post-processing step is incorporated. Experimental results show that the proposed algorithm is more accurate than the two speech rate determination algorithms on which it was inspired.

Index Terms— Speech rate determination, vowel detection

1. INTRODUCTION

Whenever speech is observed, people will have an opinion about its rate. They might believe the observed speech is slow, normal or fast. The quantification of the perceived speech rate is a topic on which research has been done already. An abstraction of a speech signal to its perceived rate could be valuable for numerous reasons. Certain speech algorithms could benefit from information of the rate of the incoming speech by adapting the algorithmic details to the given rate. One example is automatic speech recognition (ASR) [1]. Another field where speech rate is important is voice training. Professional speakers, such as people working in call centers, should produce speech at a correct rate in order to maximize intelligibility and make the listening task as agreeable as possible. Speech rate training can also benefit persons suffering from certain speech disorders. For instance, with a reduction in speech rate, significant improvements in speech intelligibility can be observed in persons with dysarthria [2].

Parts of the research reported on in this paper were performed in the context of the CATRIS (TBM-80662) project, supported by the Flemish government agency for Innovation by Science and Technology (IWT).

Multiple efforts have been made to quantify speech rate. When measuring the rate of speech, it is common practice to count the number of instances of a specific phonetic unit that are produced in a certain time span. Research has shown that the production speed of syllables is a good representation of the rhythm of speech [3]. That is why multiple speech rate determination algorithms exist that rely on counting syllables, and usually achieve this by detecting vowel positions in the acoustic speech signal. Many of these detection algorithms rely on energy or loudness measurements (e.g. [4]), sometimes complemented with verifications of periodicity (e.g. [5]). Another approach would be to use ASR to get a phonological representation of the utterance, from which the number of syllables could easily be inferred. This approach would, however, make the detection system more complex and would render it language (and dialect) specific. Furthermore, as mentioned previously, speech rate determination as a front-end processing step could increase the performance of ASR, in which case it would make no sense to rely on ASR to determine speech rate. A recognizer that simply classifies speech units into broad phonetic classes could, on the other hand, reduce the complexity and language dependence and has been utilized to measure speech rate [6]. Numerous techniques have been developed to detect vowels in an utterance or to segment speech into vowel and non-vowel like units, all of which could be used for the purpose of speech rate determination.

During previous work we compared the performance of 8 different speech rate estimators [7], many of which were based on sub-band energy envelope measurements for estimating the number of vowels. In this paper, we propose a further improvement in this class of speech rate estimation algorithms, and we called the resulting algorithm the LFME (Low Frequency Modulated Energy) algorithm. In section 2, we briefly discuss some algorithms that are based on sub-band energy envelope measurements and give the rationale for our novel approach. In section 3, we propose our LFME algo-

rithm and we finally discuss its performance in section 4.

2. BACKGROUND

One of the first publications pertaining to the automatic measurement of speech rate describes an algorithm called enrate [8]. Enrate is the first spectral moment of the energy envelope of speech and was shown to correlate with speech rate. Morgan et al. [9] later on described an enhanced method based on their enrate algorithm. They call this technique mrate. Mrate is the mean of enrate and two peak counting methods. The first peak counting is done on the energy envelope of the signal and the result is also a direct estimation of the speech rate by itself. The second peak counting is executed on the point-wise correlation $y(n)$ of the energy envelopes of 4 different band-pass signals (with band edges: 300, 800, 1500, 2500, 4000 Hz):

$$y(n) = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} e_i(n) e_j(n) \tag{1}$$

where e_i represents the (compressed) energy envelope of the i^{th} sub-band signal. $M = N(N-1)/2$ is the number of unique pairs and $N(= 4)$ is the number of sub-bands. It should be noted that the peak counting on $y(n)$ itself also represents an estimate for the number of vowels in the speech signal. In [7], we referred to this method of speech rate estimation as mpeakrate.

The Temporal Correlation and Selected Sub-band Correlation method (tcssbc) [10] is very similar to mrate, but utilizes 19 frequency bands and performs a time domain cross-correlation prior to the point-wise correlation in Eq. 1. This time domain correlation leads to a smoothing of the energy envelopes. The point-wise correlation does not involve all 19 sub-band envelopes, N sub-band envelopes are selected for this purpose. Counting the number of peaks in the resulting trajectory gives an estimate of the number of vowels in the speech signal.

In a comparative study [7] of 8 algorithms, including the above algorithms and two vowel onset detection algorithms ([11], [12]), we found that the tcssbc method outperformed the other speech rate estimation techniques.

Tcssbc is clearly inspired by the mpeakrate technique. There are two enhancements to tcssbc when compared to mpeakrate: the smoothing of the energy envelopes and the selection of sub-bands. While further investigating these two methods, we noticed indications that the smoothing step is the main reason for the superiority of the tcssbc method. Based on this insight we developed the LFME speech rate estimator, building on the mpeakrate method, in which we introduce smoother energy envelopes, an altered version of the point-wise correlation and a peak pruning post-processing step.

3. LFME ALGORITHM

3.1. Peak picking

With mpeakrate, the determination of the sub-band energy envelopes is carried out by sending the signal through a filterbank, half-wave rectifying the sub-band signals, and subsequently filtering these rectified signals with a low-pass filter. We, on the other hand, will compute the energy envelopes by means of a DFT-filterbank:

$$e_i(m) = \frac{1}{N_{fft} N_{win}} \sum_{k=k_1^i}^{k_2^i} w_k \left| X_k(m) \right|^2, \; i = 0 \ldots N-1 \tag{2}$$

where $X_k(m)$ is the Short-Time Fourier Transform (STFT) of the speech signal $x(n)$. k represents the frequency and m the time (frame) index. N_{fft} is the number of DFT points and N_{win} is the length of the analysis window, expressed in number of samples. w_k are weights that increase with increasing frequency to counteract the energy declination due to the spectral slope. $X_k(m)$ is a downsampled bandpass version of $x(n)$. Taking the squared amplitude of $X_k(m)$ gives us the energy contour of this analytic sub-band signal. To end up with an energy contour $e_i(m)$ that corresponds to a frequency bin region $[k_1^i, k_2^i]$, we sum the different energy contours $|X_k(m)|^2$ for which the frequency index k falls within this region. The resulting energy contour will have about twice the bandwidth of the analysis window, implying we can directly influence the amount of smoothing through the window type and length.

Once the energy envelopes are calculated, we utilize them to compute the following trajectory:

$$LFME(m) = e_0^2(m) \sum_{i=1}^{N-1} e_i(m) = e_0^2(m) e_s(m) \tag{3}$$

This formula is quite similar to Eq. 1. Instead of multiplying all band-combinations, however, we only multiply the lowest frequency band energy envelope with a higher frequency one. The new trajectory is an energy contour, modulated by the square of the low frequency energy, hence the name LFME (Low Frequency Modulated Energy). The rationale behind this is that vowels always possess a great deal of energy in the low frequency region. When a sound does not show enough low frequency energy content, chances are very slim it belongs to the class of vowels and the LFME will take on a low value. For a reinforcement of this effect, we squared the energy contour of the lowest band. As the digital speech signal is scaled between $[-1, 1]$, the energy contour will never exceed 1 and the squaring will lead to a non-linear compression. When the $N-1$ other frequency bands are chosen in a way that typical positions of different vowel formants

fall within one of these bands, we can safely say that vowels will cause a significant amount of energy both in e_0 and in e_s. The low frequency energy of unvoiced consonants and the high frequency energy of voiced consonants are, on the other hand, typically too low to lead to a high LFME value. Peak picking executed on this trajectory should thus give us an estimate of the number of vowels in the speech signal.

3.2. Peak pruning

As vowels introduce peaks in the LFME trajectory, we can perform peak counting on this contour to determine the number of vowels in the utterance. Of course, not all peaks correspond to a single vowel. Other sounds can produce peaks, and one vowel can generate more than one single peak. Small peaks in the trajectory are most likely undesired ones and will be discarded. This is achieved by measuring the height difference between the peak and the valley preceding it. When this difference does not exceed a set threshold, the peak is no longer considered to be an indication of a vowel position. All other peaks are kept as vowel position candidates. We will extract features at these positions to dispose of spurious peaks that remain after the peak height thresholding. Figure 1 displays a flow-chart of all peak pruning decision steps.

Some algorithms use the voiced/unvoiced classification of a PDA or zero-cross rate to exclude peaks caused by unvoiced sounds. Practically, almost no peaks caused by unvoiced sounds surpass the height threshold, as these sounds do not possess enough low frequency energy. The only exceptions are clearly pronounced plosives. To cope with this, we introduce a two-band energy ratio check. Since these problematic plosives typically show a rather flat spectrum, while vowels possess a declining spectrum, we compare the low frequency energy content to the mid and high frequency content at the peak position m_{p1}. We call the ratio of these energies $E_{r2b}(m_{p1})$. If the ratio stays lower than a certain threshold, we are probably not dealing with a vowel and the peak is discarded.

One vowel can introduce multiple peaks. This is mainly due to variations in amplitude within the vowel itself. This issue of multiple peaks can be tackled by adopting the use of a feature that describes the spectral energy distribution of speech at a particular location, but is amplitude independent. If the peak detection on the LFME trajectory gives two peaks that are located close to each other, we can extract this feature at the two peak locations and determine if the spectral distributions at the two locations are similar. If a large similarity is observed, this is a sign that the two peaks are probably caused by the same vowel and the second peak might be a spurious one.

The (amplitude independent) spectral distribution feature we use for this purpose is a normalized energy envelope vector:

$$E_n(m) = \frac{1}{\sum_{j=0}^{N-1} e_j(m)} \left(e_0(m), e_1(m), \ldots, e_{N-1}(m)\right) \tag{4}$$

The dissimilarity in spectral distribution at the two peak locations m_{p0} and m_{p1} can then be determined by computing the Euclidean distance:

$$E_{nd}(m_{p0}, m_{p1}) = \|10log_{10}E_n(m_{p1}) - 10log_{10}E_n(m_{p0})\| \tag{5}$$

When two peaks are located close together and the E_{nd} associated with the two peaks is smaller than a set threshold, the two peaks are most probably caused by the same type of vowel. It could, however, be that the two vowels are separated by another phoneme, in which case the second peak should not be discarded. To identify this scenario a second normalized energy distance $E_{nd}(m_{p1}, m_{v1})$ is extracted using the valley location m_{v1} between the two peaks, i.e. where the LFME takes on its lowest value. This allows us to track how the energy distribution varies while moving from one peak to the next and if the calculated distance $E_{nd}(m_{p1}, m_{v1})$ is large the second peak should be kept. Even if $E_{nd}(m_{p1}, m_{v1})$ is small and there is thus no considerable shift in spectral energy distribution, if there is a large rise in global signal energy when going from the valley towards the peak, the speaker most likely intended to produce two instances of the vowel (with e.g. a glottal stop in between) and both peaks should be retained nonetheless. So when $E_{nd}(m_{p1}, m_{v1})$ is small, we do one final check before we discard the second peak. We compute the signal energy at the peak and valley location and compute the ratio of these two energies $E_r(m_{p1}, m_{v1})$.

The final output of the algorithm is an estimation of the number of vowels in the speech signal. As a last step, we divide this estimated number by the length of the speech signal to arrive at the estimated speech rate in syllables/second.

4. RESULTS & DISCUSSION

In this section we elaborate on the evaluation of the proposed algorithm. For the evaluation, we used a Dutch database of read speech consisting of 648 utterances, uttered by 36 different speakers in clean conditions. All utterances in the database are between 5 and 15 syllables long. The average duration of an utterance is 2.75 seconds. Every speaker uttered a different set of sentences. The sampling rate is 16 kHz. Before we could assess the performance of the algorithm, an optimization of the parameters was required. The level of smoothing of the LFME trajectory relies on one parameter: the length of the STFT window. The peak pruning step depends on 6 threshold parameters. The other parameters (the band edges k^i and weights w_k) were given a fixed value. Comparable bands as the ones utilized by the mpeakrate

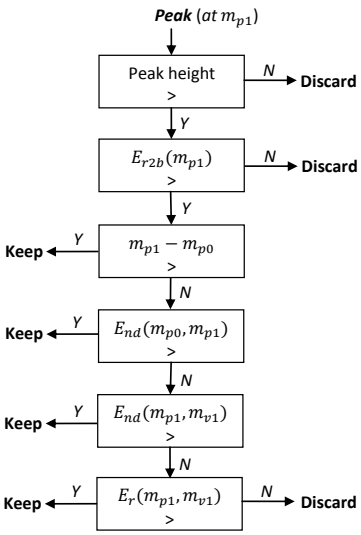

Fig. 1. Peak pruning schematic

algorithm were used for the LFME calculation: e_0 corresponded to the energy contained in the band $(50, 800)$Hz, while e_s reflected the energy in the $(800, 4000)$Hz region. This band was further split up into 3 regions with band edges 800, 1500, 2500 and 4000Hz for the calculation of E_{nd}. The weights w_k were chosen to counterbalance a spectral slope of -6dB/octave and as such obtain a flattened spectrum. A Hann analysis window was used for the calculation of the STFT. We used half of the speakers (13) in the database for the optimization, resulting in 324 utterances (The remaining 13 speakers were used for the evaluation of the algorithm). The two most influential parameters that were optimized are the window length and the peak height threshold. We optimized these parameters separately from the other 5 peak pruning threshold parameters. We first used an automatic speech recognizer in forced alignment mode to identify the vowel borders. ASR is, however, not flawless, so the number of vowels in each utterance was also manually determined by a human listener and this reference is used to optimize the two most important parameters. This reference number is not perfect, as human listeners tend to overlook added sounds (mostly 'schwa') and sometimes do perceive sounds that are absent, but are expected to be there. Still, we believe it was much wiser to use this manual reference to optimize the two most important parameters, as the vowel borders given by ASR would not be reliable enough and, additionally, forced alignment does not account for any unintentionally added or omitted vowels in the utterance.

We first computed the LFME trajectories using a specific window length, identified all peaks and only retained those that were higher than a certain peak height threshold. At the positions corresponding to these peaks, we extracted the 5 remaining peak pruning features and the type of peak; the peak was either a desired peak or a spurious peak. A spurious peak is a peak caused by a non-vowel sound or the non-first peak occurring within a vowel segment. All peaks that occurred first in a vowel segment, were classified as desired peaks. The vowel borders that were used for this purpose are the ones as identified by ASR.

The 5 peak pruning thresholds were optimized using a pattern-search algorithm in such a way that the sum of the number of retained spurious peaks and the number of discarded desired peaks is minimized. We then used these thresholds to optimize the window length and peak height threshold. We ran the full algorithm (including all peak pruning steps) on the optimization utterances using a fixed grid of window length and peak height threshold combinations, each giving an estimated number of syllables. We then searched for the combination that minimized the RMS error:

$$E_{RMS} = \sqrt{\frac{1}{N_{utt}} \sum_{u=1}^{N_{utt}} \left(\frac{syll_d(u) - syll_r(u)}{syll_r(u)} \right)^2} \quad (6)$$

where $syll_d$ is the number of detected syllables, $syll_r$ is the reference number of syllables as determined by the human listener, u is the utterance number and N_{utt} is the number of utterances in the optimization database. Minimizing this RMS error does not optimize the accuracy of vowel detection, rather it optimizes the detection of the number of vowels in the whole utterance. The error of missed peaks that are compensated for by the detection of false peaks is not considered. We still chose to use Eq. 6 since we wanted to base the optimization of the most influential parameters on the manual determination of the number of vowels and not on ASR. Additionally, we would like to note that the LFME algorithm as we use it, is intended for speech rate determination, and that a speech rate measurement is not influenced by a compensation of missed vowels by falsely detected vowels.

After the optimization of the window length and height threshold, the whole process is repeated, starting with computing new LFME trajectories with the optimized window length and rejecting peaks according to the optimized peak height threshold, after which the peak pruning thresholds, and subsequently, the window length and peak height threshold were re-optimized. We continued this process until stable optimized parameter values were obtained.

We compared the performance of the proposed LFME algorithm to that of the mpeakrate part of the mrate algorithm, and to the performance of the tcssbc algorithm. These other 2 methods also involve counting the number of peaks in a feature trajectory. The peak height threshold that determined which peaks to keep and the smoothing parameter used in the tcssbc method were optimized in the same way as done for the LFME method, i.e. by determining the parameter combination that minimized the RMS error of Eq. 6.

	E_{RMS}	$corr$
$tcssbc$	0.165	0.75
$mpeakrate$	0.279	0.49
$mpeakrate\_dft$	0.162	0.77
$LFME\_base$	0.143	0.81
$LFME$	0.132	0.83

Table 1. Experimental results

The dissimilarities between the proposed method and mpeakrate lie in the smoothing of the energy contours, the calculation of the feature trajectory and the post-processing peak pruning. In order to systematically investigate the influence of each of these aspects, we also look at two methods that are a combination of these two techniques: we also consider the mpeakrate method where we used a DFT to calculate smoothed energy envelopes using an optimized window length (mpeakrate_dft) and we investigate the LFME method with (LFME) and without (LFME_base) the 5 last peak pruning steps.

Table 1 shows for every considered method the RMS error on the number of detected syllables, and the correlation coefficient between the calculated speech rate and the reference speech rate. For the calculations of these figures we used the remaining 13 speakers (324 utterances) that were not used during optimization. When using the same energy envelope sampling rate, LFME typically took about 40% and 70% of the computation times required by tcssbc and mpeakrate respectively. We can see that as we move from mpeakrate to LFME, by first introducing smoothing, then the LFME trajectory and, finally, additional peak pruning, with every step the RMS error reduces and the correlation coefficient increases. From Table 1 it is also clear that if we introduce smoothing, mpeakrate performs comparable to tcssbc, indicating that smoothing is the main reason for the superiority of the tcssbc trajectory, and not so much the selection of sub-bands.

5. CONCLUSION

In this paper we introduced the LFME algorithm. The conducted experiments showed that peak picking performed on the LFME trajectory results in a better speech rate estimation than when it is done on the mpeakrate or tcssbc trajectories. The introduction of additional peak pruning can further enhance the estimation process. The LFME algorithm not only outperforms the tcssbc algorithm, which from a previous study was believed to be the best performing technique, but is in addition less demanding in terms of computational complexity. Even though we only performed experiments using Dutch speech, we expect the algorithm to also work with other Western European languages, which share the same syllabic structure as Dutch, since energy based vowel detection is generally believed to be largely language independent.

REFERENCES

[1] Matthew Richardson, M Hwang, Alex Acero, and Xuedong Huang, "Improvements on speech recognition for fast talkers.," in *Eurospeech*, 1999.

[2] Kathryn M Yorkston, Vicki L Hammen, David R Beukelman, and Charlie D Traynor, "The effect of rate control on the intelligibility and naturalness of dysarthric speech," *Journal of Speech and Hearing Disorders*, vol. 55, no. 3, pp. 550, 1990.

[3] Sieb Nooteboom, "The prosody of speech: melody and rhythm," *The handbook of phonetic sciences*, pp. 640–673, 1997.

[4] Thilo Pfau and Günther Ruske, "Estimating the speaking rate by vowel detection," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 1998, vol. 2, pp. 945–948.

[5] Nivja H de Jong and Ton Wempe, "Praat script to detect syllable nuclei and measure speech rate automatically," *Behavior research methods*, vol. 41, no. 2, pp. 385–390, 2009.

[6] Jiahong Yuan and Mark Liberman, "Robust speaking rate estimation using broad phonetic class recognition," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 4222–4225.

[7] Tomas Dekens, Mike Demol, Werner Verhelst, and Piet Verhoeve, "A comparative study of speech rate estimation techniques," in *INTERSPEECH*, 2007, pp. 510–513.

[8] Nelson Morgan, Eric Fosler-Lussier, and Nikki Mirghafori, "Speech recognition using on-line estimation of speaking rate," in *Eurospeech*, 1997, vol. 97, pp. 2079–2082.

[9] Nelson Morgan and Eric Fosler-Lussier, "Combining multiple estimators of speaking rate," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 1998, vol. 2, pp. 729–732.

[10] Shrikanth Narayanan and Dagen Wang, "Speech rate estimation via temporal correlation and selected subband correlation," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2005, pp. 413–416.

[11] Dik J Hermes, "Vowel-onset detection," *The Journal of the Acoustical Society of America*, vol. 87, pp. 866, 1990.

[12] SR Mahadeva Prasanna, Jinu Mariam Zachariah, and B Yegnanarayana, "Begin-end detection using vowel onset points," in *Workshop on Spoken Language Processing*, 2003.