INFLUENCE OF SILENCE REMOVAL ON SPEAKER RECOGNITION BASED ON SHORT POLISH SEQUENCES

Adam Dąbrowski, Tomasz Marciniak, Agnieszka Krzykowska, Radosław Weychan

Poznań University of Technology, Chair of Control and Systems Engineering,
Division of Signal Processing and Electronic Systems,
ul. Piotrowo 3a, 60-965 Poznań, Poland,
e-mail: tomasz.marciniak@put.poznan.pl

Abstract: This paper studies effectiveness of speaker identification based on short Polish sequences. The results have been got as a continuation of the experiments presented by the authors during the previous SPA conference. An impact of automatic removal of silence on the speaker recognition accuracy is considered. Several methods to detect the beginnings and ends of the words have been used. Experimental studies have been conducted in Matlab for a specially prepared database of short speech sequences in Polish.

1 Introduction

As mentioned in our previous article [1], techniques based on acoustic signals are an interesting solution in some biometry applications. In our present study we focus on experiments with short speech sentences. In conclusion, the article states that improving the efficiency of the speaker identification can be possible after removing unnecessary speech parts, e.g. due to application of precise end-point detection (EPD) algorithms.

This article examines an influence of the voice activity detection techniques on efficiency of the speaker identification, which has been realized using the GMM (Gaussian mixture models) as well as VQ (vector quantization) algorithms. The article is organized as follows: Section 2 presents the used EPD methods, Section 3 briefly describes the speaker identification techniques. Chapter 4 includes the results of experiments in a form of the FAR / FRR graphs, while section 5 summarizes our work.

2 EPD methods

Exact detection of voice activity endpoints is crucial in many speech processing procedures like eg.: speech coding in telephone communication, speech enhancement, automatic speech or speaker recognition [2].

In case of automatic speech / speaker recognition, the precise detection of word boundaries can be a crucial step, which significantly improve the recognition effectiveness. Voice activity detection is realized in time and/or in frequency domain using, e.g., the TF (time-frequency) parameters speech parameters.

Detection and removal of silence was made for the two solutions. In the first part of the experiment were removed only silence at the beginning and end of the speech sequence. In the next stage of the study is the detection of individual words in the analyzed sentence.

Operation of methods is illustrated by using an illustrative wave sequence „Chciałbym zgłosić wypadek" ("I would like to report an accident"). This wave file was recorded in an anechoic chamber with the use of the Matlab environment with the sampling rate of 22 050 frames per second. Table 1 shows the applied algorithms prepared by Roger Jang [3].

Table 1. Applied EPD algorithms

Method	Short Description
Energy analysis	Calculation of energy value
Jang (v2)	Application of volume threshold
Jang HOD	Use of volume and high-order differences
Jang ZCR	Silence detection based on volume and zero-crossing rate

2.1 Energy analysis

Simplest method of EPD is analysis of signal energy:

$$E_i = \sum_{n=k_{ip}}^{k_{ik}} x^2(n) \tag{1}$$

where i stands for the number of the window of the signal x, k_{ip} is the first sample of the signal, and k_{ik} is the last.
Length of the window used to count the energy is: 0,01 [ms]. Number of samples in one window is 220, and the offset of the window is equal to 0,001 [ms]. Example result of wave processing using this algorithm is shown in Fig. 1.

2.2 Jang algorithm (v2)

The algorithm finds the beginning and the end of the speech samples based on the volume threshold V_t defined as:

$$V_t = \frac{V_{\max} - V_{\min}}{V_r} + V_{\min} \tag{2}$$

where coefficient V_r is 10, V_{min} i V_{max} are minimal and maximal value of volume vector, which every element is computed from the equation:

$$V_i = \sum_{n=k_{ip}}^{k_{ik}} |x(n)| \tag{3}$$

In equation (3) i means number of windows used to count the volume, and k_{ip} and k_{ik} are the first and the last samples. The length of the window is: 0,016 [ms]. Number of samples: 352. Window offset: 0 [ms].
Authors of these algorithms comes to a conclusion that using a constant volume threshold does not allow to get good results if the volume of the signal varies.

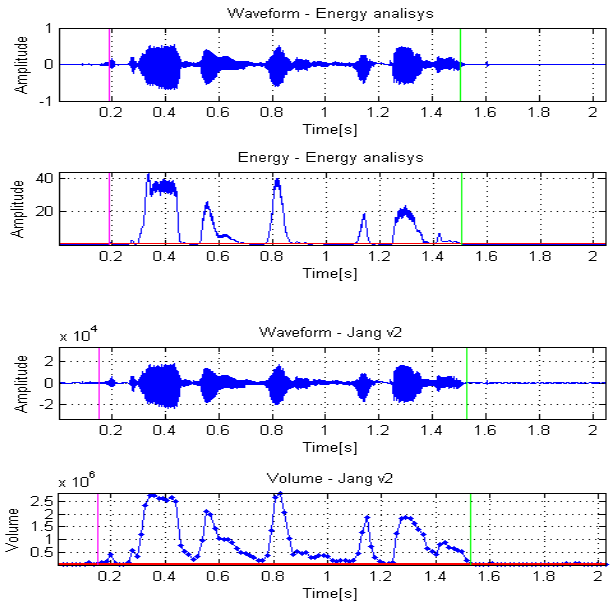

Fig. 1. Detection of beginning and end of the sentence using Energy analysis algorithm and Jang (v2) algorithm

2.3 Jang HOD algorithm

This algorithm uses high-order differences of the given signal as characteristics in time domain. Individual steps of the algorithm are as follows:
1. Computing volume (V) as in Jang (v2) using equation (3) and absolute value of the sum of j-order difference (H).

$$H_i = \sum_{n=k_{ip}}^{k_{ik}} \left| \frac{\Delta^j x(n)}{\Delta n^j} \right| \tag{4}$$

where i stands for number of the time window used to compute H_i. k_{ip} and k_{ik} are the first and the last sample of given window. Before moving to next step of the algorithm values of V and H are normalized.

2. Selection weights w from interval [0, 1] to compute new curve VH:

$$VH = w \times V + (1-w) \times H \tag{5}$$

3. Finding coefficient r to compute threshold t for VH in order to determine end-points. The threshold is defined as:

$$t = VH_{\min} + (VH_{\max} - VH_{\min}) \times r \tag{6}$$

Default values of parameters are: j = 4, w = 0.5, and r = 0.125. Length of the time window to compute volume is: 0,016 [ms]. Number of samples in one window is: 352. Overlapping of windows is 0 [ms]. Fig. 2 presents an illustration of Jang HOD algorithm.

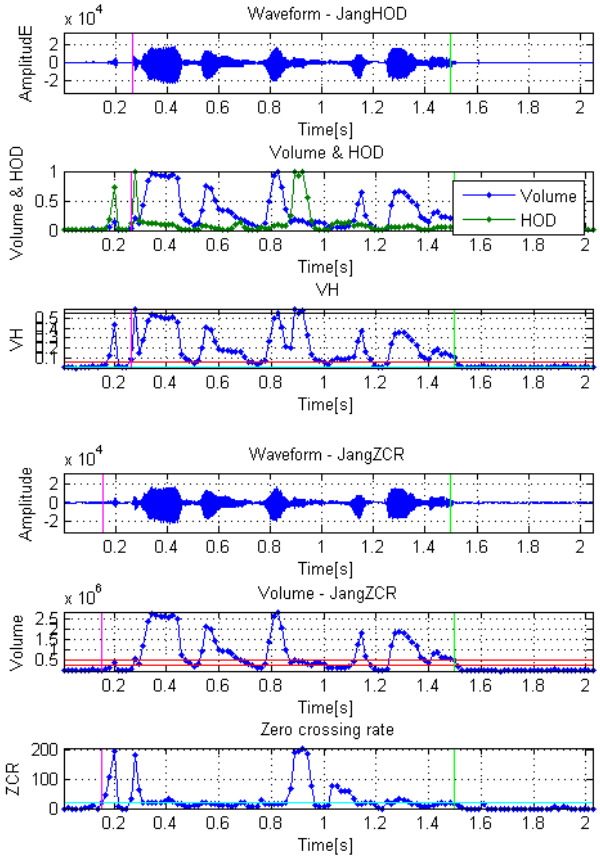

Fig. 2. Detection of start and end of the sentence using Jang HOD and Jang ZCR algorithms

2.4 Jang ZCR algorithm

This method defines the beginning and the end of words on the basis of the volume threshold and zero crossing rate. Steps of the algorithm:
1. Selection initial beginning and ending points based on the threshold τ_u.
2. Expanding borders to cross the threshold τ_l.
3. Calculation of Z (ZCR) coefficient and further expanding of range to cross threshold τ_{zc} defined as:

$$\tau_{zc} = \max(Z) \times r_z \tag{7}$$

where r_z = 0.1, and Z (ZCR) is vector of elements, in which each one is defined as:

$$Z_i = \sum_{n=k_{ip}}^{k_{ik}-1} I\{x(n)x(n-1) < 0\} \qquad (8)$$

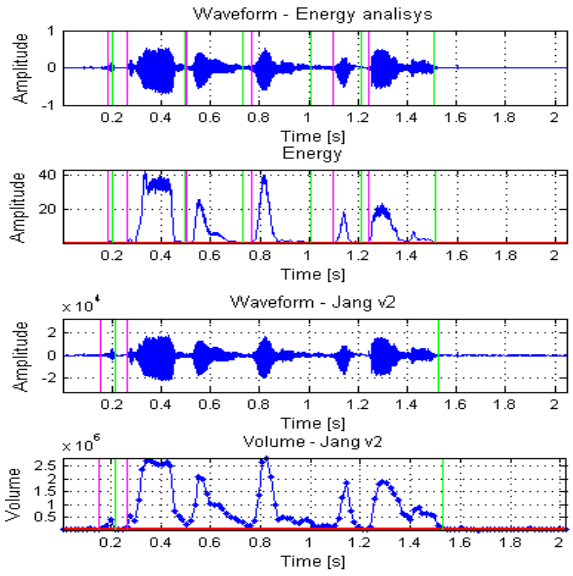

Fig. 3. Detection of silence at the beginning, in the middle, and at the end of sentence using energy analysis and Jang (v2) algorithm

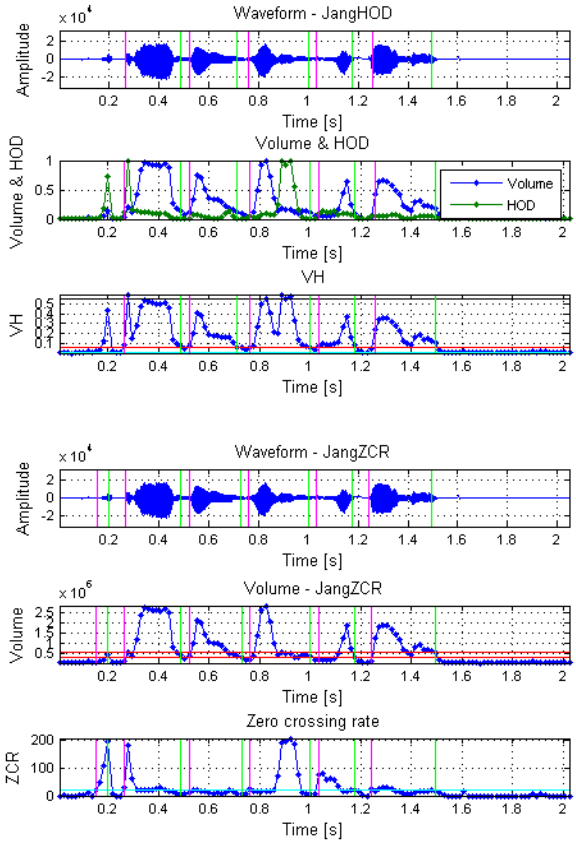

Fig. 4. Detection of silence at the beginning, in the middle, and at the end of sentence using Jang HOD and Jang ZCR algorithms

Length of the time window used to computer volume is 0,016 [ms]. Number of samples of the time window is 352, where overlapping: 0 [ms]. Example results of wave processing using this algorithm is shown in Fig.3.

2.5 Modification of Jang functions

During the experiment also second approach was tested. Its main goal was to analyze what impact would have cutting silence not only on the beginning and end of the sentence but also in the middle i.e. between the words (cf., Fig. 4). Because tested signals are very short, there are not many words in the sequence. Therefore not much of silence can be cut. Nevertheless in some cases this approach shown improvement relation to previous one.

3 Speaker recognition algorithms

3.1 Feature extraction

Speech signal was parameterized using MFCC (*mel frequency cepstral coefficients*). It uses a fact of logarithmic sound perception of human's ears. Speech is divided into frames being equivalent to about 23,2 ms in time domain. This frames are multiplied by Hamming window function. From every frame DFT coefficient are computed and translated to Mel scale. Finally, MFCC are calculated.

3.2 Vector quantization

Vector quantization is an algorithm that models using little amount of representative feature vectors to build the codebook. Main parameter controlling speed of algorithm is amount of centroids, which was set in this case to 32.

Quantized MFCC coefficients are generated using filter bank containing 30 filters. Low cutoff frequency of first filter is 0 Hz, and high cutoff for last filter is 8000 Hz.

During our research Matlab functions from the project [4] and VOICEBOX [5] were used. Used software was supplemented for a batch processing, non-deterministic model selection and statistics generation.

3.3 GMM

Gaussian Mixture Models is one of the most commonly used algorithms to create speaker model in speaker recognition system. First, speaker voice is modelled by MFCC. Then obtained feature vectors are used to train the model with GMM. Expectation-maximization (EM) algorithm is used during this step. As described in our previous paper [1] our software is based on the approach presented at web page [6].

4 Experimental results

Experiments was performed in few stages. Speech database was processed with described silence removal algorithms giving 9 independent bases for research of influence of silence removal on speaker recognition accuracy using VQ and GMM algorithms. Experiment result is multi-dimensional matrix containing coefficients of similarity between processed samples and models, for every sentence. Using this matrix FAR / FRR curve is computed using DETware [7].

Because in our recordings average time of speech is 1 second we combined features from 5 random speech sequences to obtain sufficient speaker model.

4.1 Database of short Polish sentences

During experiments, a database of 40 speakers of both sex and age 20 to 55 was prepared. Each of speaker speak 30 times 6 sentences:

- Dzień dobry (Good morning)
- Do widzenia (Good bye)
- Dobry wieczór (Good evening)
- Moje nazwisko (My name)
- Chciałbym zgłosić wypadek (I would like to report an accident)
- Czas nadziei nie trwa wiecznie (Time of hope doesn't last forever)

Recordings was realized in three stages. Every speaker repeated every sentence 10 times at ones. Time period between sessions was 1 to 6 weeks.

Every of 7200 samples was recorded in anechoic chamber using omnidirectional characteristic condenser microphone. Sampling frequency of recorded samples was set to 22050 Hz and 16 bit resolution.

4.2 Results after detection start and end of the sentence

Vector quantization

Fig. 5 presents FAR / FRR plots for unprocessed (raw) and processed with silence removal speech. In this case, only silence at the beginning and at the end is removed. It can be seen, that silence removal algorithms improve speaker recognition accuracy. Three of presented algorithms give similar results (energy, Jang HOD and Jang ZCR) and EER takes place on about 7-7,5 % for them. In case of unprocessed speech, this coefficient is about 10 % higher.

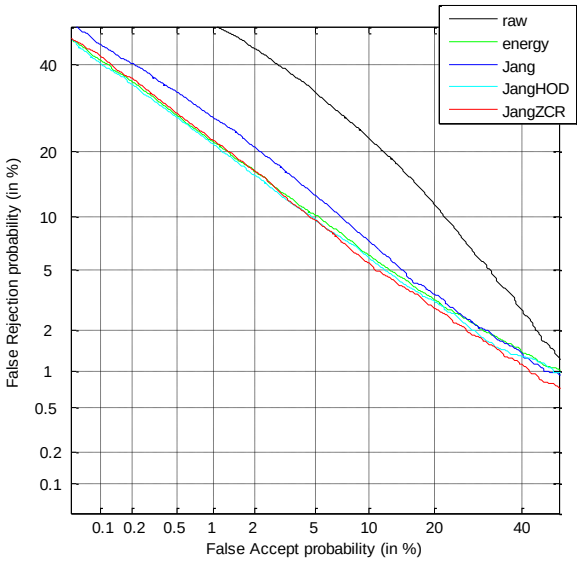

Fig. 5. FAR / FRR plots for basic algorithms of speech removal for VQ processing

GMM

Fig. 6 shows FAR / FRR plots of speaker recognition with usage of 5 databases containing raw (unprocessed) speech and wavelet with removed silence at the beginning and at the end of sequence. It can be inferred that speaker recognition can be improved by silence removal algorithms, though each algorithm gives similar effect and their EER is about 7 %. For unprocessed speech EER is about 11 % for GMM processing.

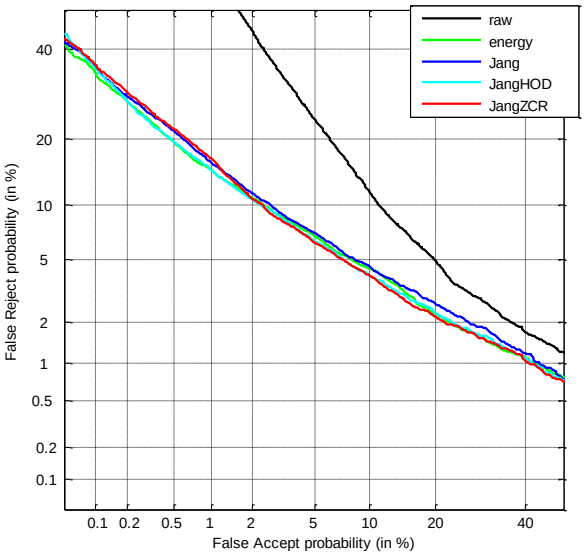

Fig. 6. FAR / FRR plots for basic algorithms of speech removal for GMM processing

4.3 Results after removing silence between words

Vector quantization

In this case, silence was removed from all of recorded speech, e.g. between spoken words. Fig. 7 presents FAR / FRR plots for raw and processed speech. It can be noticed, that enhanced Jang ZCR algorithm do not trend to improve. Little correction can be seen for enhanced Jang algorithm. Greatest improvement received enhanced energy and JangHOD algorithm. In this case, EER lowered to about 5 %.

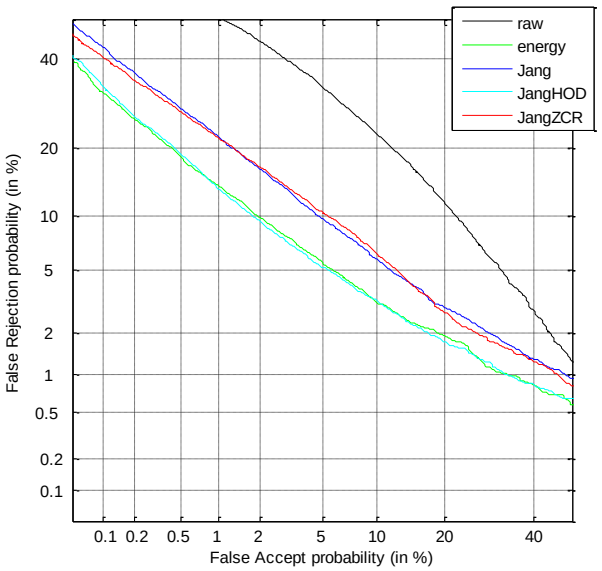

Fig. 7. FAR / FRR plots for enhanced algorithms of speech removal

GMM

Fig. 8 shows FAR / FRR plots of speaker recognition with usage of raw speech and speech with removed silence at the beginning in the middle and at the end of sequence. In this case speaker recognition is also improved by silence removal algorithms, and each algorithm gives similar effect. EER for databases with removed silence is about 6-7 %. For unprocessed speech EER is about 11 %.

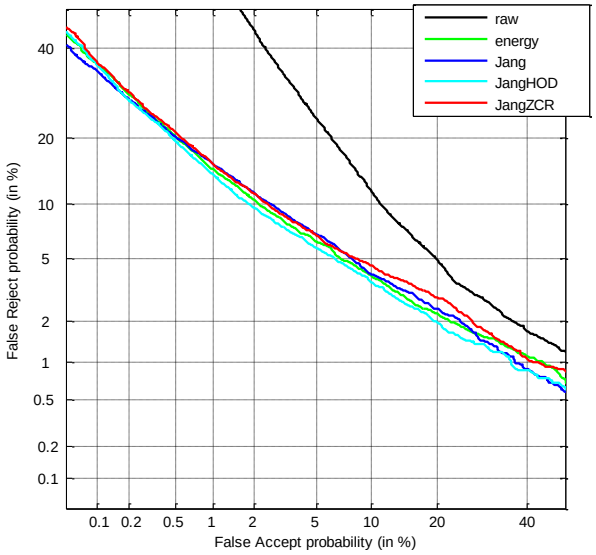

Fig. 8. FAR / FRR plots for enhanced algorithms of speech removal using GMM processing

5 Conclusions

As expected, the removal of unnecessary sections of the recording (i.e. the silence), resulted in an improved speaker identification performance of about 5 %. The results obtained for different methods are quite similar, but the Jang HOD method can be distinguished. Note that the significantly noisy signal (e.g. during the telephone transmissions) may be more sensitive to the choice of the EPD method.

It seems that further improvement should be sought in the adaptive selection of the length of the analyzed blocks of samples. The authors also plan to investigate techniques based VAD algorithms in time and frequency with the generalized autoregressive conditional heteroscedasticity (GARCH).

References

[1] Marciniak, T., Weychan, R., Drgas, Sz., Dąbrowski, A., Krzykowska, A., Speaker recognition based on short polish sequences, Proc. of SIGNAL PROCESSING SPA'2010, Poland Section, Chapter Circuits and Systems IEEE, pp. 95-98, Poznań, Poland, September, 23-25th 2010.

[2] Marciniak, T., Dąbrowski, A., Rochówniak, R., Subband wavelet signal denoising for voice activity detection, *Proc. of NTAV/SPA '2008*, pp. 93-96, Poznań, Poland, September, 25-27, 2008.

[3] Jyh-Shing Roger Jang, "ASR (Automatic Speech Recognition) Toolbox", available from the link at the author's homepage at http://mirlab.org/jang

[4] DSP Mini-Project: An Automatic Speaker Recognition System http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition

[5] Voicebox: speech processing toolbox for Matlab http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[6] A. Alexander, A. Drygajlo Speaker identification: A demonstration using Matlab, http://scgwww.epfl.ch/matlab/student_labs/2005/labs/

[7] DET-Curve Plotting software for use with MATLAB, http://nist.gov/itl/iad/mig/tools.cfm

This work was partly supported by INDECT, PPBW, and DS 2011 projects.