DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **17**, No 4
Sofia • 2017 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2017-0045

LTSD and GDMD features for Telephone Speech Endpoint Detection

Atanas Ouzounov

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
E-mail: atanas@iinf.bas.bg

Abstract: *This paper proposes a new contour-based speech endpoint detector which combines the log-Group Delay Mean-Delta* (*log-GDMD*) *feature, an adaptive two-threshold scheme and an eight-state automaton. The adaptive thresholds scheme uses two pairs of thresholds – for the starting and for the ending points, respectively. Each pair of thresholds is calculated by using the contour characteristics in the corresponded region of the utterance. The experimental results have shown that the proposed detector demonstrates better performance compared to the Long-Term Spectral Divergence* (*LTSD*) *one in terms of endpoint accuracy. Additional fixed-text speaker verification tests with short phrases of telephone speech based on the Dynamic Time Warping* (*DTW*) *and left-to-right Hidden Markov Model* (*HMM*) *frameworks confirm the improvements of the verification rate due to the better endpoint accuracy.*

Keywords: *endpoint detection, long-term spectral divergence, group delay spectrum.*

1. Introduction

The aim of the Endpoint Detection (ED) is to locate the starting and the ending points of a speech utterance. This detection is a crucial preprocessing stage in automatic speech and speaker recognition systems designed to operate in noisy real-world environments.

In the endpoint detection there are typically two main processing steps - feature extraction and decision scheme. In the first processing step, the features based on signal energy [17, 18], autocorrelation functions [42], spectral entropy [13, 36, 39], time-frequency parameters [15], wavelets [38, 39], bi-spectrum [20], etc., are extracted. In the second step, by using the properties of the estimated features, the starting and the ending points of the utterance are estimated. This is accomplished by applying state automaton [18, 37] or a classification scheme based on Hidden Markov Models (HMMs) [41], support vectors machines [29], neural networks [35], etc.

The effectiveness of four explicit contour-based single-feature speech endpoint detection algorithms have been evaluated experimentally in the study. The first algorithm adopts as a feature the log version of the described in [24] the Group Delay Mean-Delta (GDMD) feature (named as log-GDMD). As a decision scheme is applied the proposed in the study a new adaptive two-threshold scheme and a new eight-state automaton and this detector is denoted as GDMD-E. The second algorithm is the well-known Long-Term Spectral Divergence (LTSD) algorithm [28] with hangover scheme [8] and it is denoted as LTSD-H. This algorithm is usually used in the Voice Activity Detection (VAD) tasks, but in this research its endpoint detection capability is evaluated. The third and fourth algorithms are new and are designed as a combination of features and decision schemes from the previous two ones. The third algorithm is a combination of the log-GDMD feature and the hangover scheme [8] and is denoted as GDMD-H. The last algorithm is a combination of the LTSD feature and proposed in the study algorithm with adaptive two-threshold scheme and eight-state automaton and it is denoted as LTSD-E.

In order to validate the performance of the four endpoint detection algorithms, two experiments are carried out. In the first one the accuracy was evaluated in terms of frames differences between hand-labelled and detected endpoints. In the second experiment the performance of the endpoints detection algorithms in terms of the recognition rate is estimated via two fixed-text speaker verification applications. The first application is based on the Dynamic Time Warping (DTW) algorithm [21] while the second one uses the left-to-right HMM paradigm [10]. The verification results are compared to these obtained by the manual endpoint detection. Two speech corpora are used in the experiments – TDIGITS corpus (in English) [6] and BG-SRDat corpus (in Bulgarian) [23]. The first experiment uses noisy speech examples from both corpora, while the second one – only examples from the latter.

The Z_{HTER} – test method proposed in [3] is used to assess the difference (in statistical sense) between the endpoint detectors by using the obtained verification rate. To illustrate the verification results the Detection Error Tradeoff (DET) curves are plotted [19].

This paper is organized as follows: Section 2 presents endpoint detection features. Section 3 describes the proposed endpoint detection algorithm with adaptive two-threshold scheme and eight-state automaton. Section 4 presents the proposed endpoint detectors. Section 5 describes experimentation with the proposed detectors. Section 6 summarizes the work and suggests future research directions.

2. Endpoint detection features

2.1. The log-GDMD feature

The GDMD feature is proposed by the author in [24]. The core idea of the GDMD feature is to use the mean value of the delta spectral autocorrelation function defined on the Modified Group Delay Spectrum (MGDS) [12] as feature in speech detection. Thus by using the MGDS, peak-enhanced delta spectral autocorrelation function is obtained and thereafter more effective mean delta feature. In this study the

log-GDMD feature is proposed which is obtained from the GDMD one by applying additional processing steps. In the text below the index *n* is omitted whenever this dependence is clear from the context. For *n*-th speech frame, $n = 0,..., N-1$, *N* is the contour's frames number, the log-GDMD feature is computed into three steps, as follows:

Step 1. Calculation of the MGDS according to [12]

- Let $x(i)$ is the given speech frame, $i = 0,..., I-1$, *I* is the number of samples in the frame.
- Compute the Fast Fourier Transform (FFT) with size *K* of the sequences $x(i)$ and $ix(i)$. Let these transforms are $X(k)$ and $Y(k)$, respectively.
- Compute the $S(k)$-cepstrally smoothed spectrum of $|X(k)|$ using low-order cepstral lifter l_w.
- Compute the MGDS $\tau_m(k)$ as

$$\tau_m(k) = \text{sign}\left|\frac{X_R(k)Y_R(k) + Y_{\text{Im}}(k)X_{\text{Im}}(k)}{S(k)^{2\gamma}}\right|^{\alpha}, \quad (1)$$

where "sign" is the sign of the term

$$\frac{X_R(k)Y_R(k) + Y_{\text{Im}}(k)X_{\text{Im}}(k)}{S(k)^{2\gamma}}. \quad (2)$$

- Parameters α, γ and l_w are adjusted according to the particular requirements.

Step 2. Calculation of the log-MD feature, using the MGDS $\tau_m(k)$ (1)

- Compute the average MGDS – averaged over all frames in the utterance.
- Obtain the mean normalized MGDS $\tau_m^n(k)$ by dividing the frame MGDS by the average MGDS.
- Compute the non-normalized unbiased spectral autocorrelation function $R_m(l)$ using the mean normalized MGDS $\tau_m^n(k)$

$$R_m(l) = \frac{1}{K/2 - l}\sum_{k=0}^{K/2-l}\tau_m^n(k)\tau_m^n(k+l), \quad (3)$$

where *K* is the number of points in the discrete Fourier transform, $l = 0,..., L$, *L* is the number of correlation lags, and $L = K/4$.

- Compute the delta spectral autocorrelation function $\Delta R_m(l)$ using $R_m(l)$ with delta window *Q* as

$$\Delta R_m(l) = \frac{\sum_{q=-Q}^{Q} qR_m(l+q)}{\sum_{q=-Q}^{Q} q^2}. \quad (4)$$

- Perform a contour smoothing for delta spectral autocorrelation function $\Delta R_m(l)$ by using *J*-order long-term spectral envelope algorithm [28]. The obtained smoothed version of $\Delta R_m(l)$ is denoted as $\Delta R_m^S(l)$,

(5) $$\Delta R_m^S(n,l) = \max\left\{\Delta R_m(n+j,l)\right\}_{j=-J}^{j=+J}.$$

- Compute the log-GDMD m_{gd} using $\Delta R_m^S(l)$ as

(6) $$m_{gd} = \log\left[\sum_{l=0}^{L}\left|\Delta R_m^S(l)\right|\right].$$

Step 3. Perform the contour smoothing and normalization

- Perform the m_{gd} contour smoothing by moving average filter with length of 5 frames.
- Compute the contour normalized log-GDMD $m_{gd}^*(n)$ as

(7) $$m_{gd}^*(n) = \left|m_{gd}(n) - m_{gd}^{\min}\right|,$$

where $m_{gd}^{\min} = \min_n\{m_{gd}(n)\}$.

2.2. The LTSD feature

The LTSD is proposed in [28]. It is defined as the deviation of the long-term spectral envelope respect to the average noise spectrum. The LTSD is utilized as feature in the VAD algorithms [16, 28]. The LTSD version described in [16, §2] will be used in the study. It has been chosen because in the preliminary endpoint detection accuracy tests it gave better results than the version in [28].

The hangover technique published by the European Telecommunications Standard Institute (ETSI) in [8] is applied in the work. This technique is used because it is described in details [1, 8] and has shown good results in the preliminary tests.

The LTSD-VAD algorithm is here evaluated in the endpoint detection task. In this case the endpoint detection is done as the first and last frames in the utterance labelled as speech are accepted as actual endpoints.

3. Endpoint detection algorithm

The proposed Endpoint Detection (ED) algorithm is designed to detect the starting and ending points of the word or short phrase. It is based on the analysis of the single-feature contour variations and uses adaptive two-threshold scheme and state automaton to make a decision.

3.1. Thresholds' setting

The transition from unvoiced to voiced part at the beginning of the utterance results in large increase in the contour's values (e.g., speech energy). On the other hand, the transition from voiced to unvoiced part in the end of utterance often causes a slow reduction in these values. Such variations in the contour make the fixed thresholds scheme difficult to set reliable thresholds. This is especially true for the unvoiced frames at the end of the utterance which are more likely to be misclassified as noise.

To reduce the errors in these cases a threshold setting algorithm which uses two pairs of thresholds is proposed in the study. The first pair is intended for detection of

the starting point, while the second one – for the ending point. Or in other words, two thresholds – low and high – are set using the contour characteristics in the beginning part of the speech record and these thresholds are used by state automaton for starting point detection only. And in similar way the two other thresholds – low and high – are set using the contour specifics in the ending part and they are used only for detection of the ending point. The key issue in this algorithm is how to define the beginning and ending parts in speech record based only on the contour features. In order to do this, it is proposed to use the contour peaks analysis.

Let $P = \{p_i\}, i = 1,\ldots,G,$ is the set of peaks, where G is the total number of peaks in analyzed contour. Each peak is defined as $p_i = (v_i, l_i)$ where v_i is the peak value and l_i is the location of the peak, i.e., the number of contour frame where the peak is placed. Let define new set $Q_M = \underset{v}{\text{sort}}\{P\}$ obtained after sorting the peaks over the peak values v_i in descending order and select the first M of them and $M<<G$. Let define $l_{\min}$ and $l_{\max}$ where $l_{\min} = \min_{l}\{Q_M\}$ and $l_{\max} = \max_{l}\{Q_M\}$. The position of the splitting point l_{spl}, i.e., the point that divide the contour into two parts – beginning and ending – is defined as $l_{\text{spl}} = l_{\min} + \kappa(l_{\max} - l_{\min})$.

In the proposed algorithm for each part of the contour a single initial threshold is computed. By using of this threshold two additional averages m_{down} and m_{up} are estimated. The first average is calculated from the contour values smaller than the threshold, while the second one – from the values equal to or greater than it (a similar idea for VAD with two averages computed from the speech energy by using a single threshold is described in [14]). Using these averages, the pairs of thresholds $T_{\text{beg}}^{\text{low}}, T_{\text{beg}}^{\text{high}}$ for beginning part of the contour and $T_{\text{end}}^{\text{low}}, T_{\text{end}}^{\text{high}}$ for the ending one are defined. The thresholding algorithm is as follows.

Step 1. Compute the contour values $E(n) \geq 0; n = 1,\ldots N$, N is number of frames.

Step 2. Find contour peaks $P = \{p_i\};\ p_i = (v_i, l_i);$ v_i is the peak value, l_i is the location of the peak and $i = 1,\ldots,G$, G is the number of peaks.

Step 3. Find $Q_M = \underset{v}{\text{sort}}\{P\}$ in descending order and select the first M peaks; $M<<G$.

Step 4. Compute $l_{\min} = \min_{l}\{Q_M\}$ and $l_{\max} = \max_{l}\{Q_M\}$.

Step 5. Compute splitting point $l_{\text{spl}} = l_{\min} + \kappa(l_{\max} - l_{\min})$.

Step 6. Compute initial thresholds for beginning and ending part of the contour:

$$\begin{aligned} T_{\text{beg}}^{\text{init}} &= \text{E}\{E(n)\};\ \ n = 1,\ldots,l_{\text{spl}}, \\ T_{\text{end}}^{\text{init}} &= \text{E}\{E(n)\};\ \ n = l_{\text{spl}}+1,\ldots,N. \end{aligned} \tag{8}$$

Step 7. Compute additional average values for the beginning part:

(9) $$m_{\text{beg}}^{\text{down}} = \frac{\sum_{n=1}^{l_{\text{spl}}} E(n)w(n)}{\sum_{n=1}^{l_{\text{spl}}} w(n)},\ w(n) = \begin{cases} 1 & \text{if } E(n) < T_{\text{beg}}^{\text{init}}, \\ 0 & \text{otherwise,} \end{cases}$$

(10) $$m_{\text{beg}}^{\text{up}} = \frac{\sum_{n=1}^{l_{\text{spl}}} E(n)v(n)}{\sum_{n=1}^{l_{\text{spl}}} v(n)},\ v(n) = \begin{cases} 1 & \text{if } E(n) \geq T_{\text{beg}}^{\text{init}}, \\ 0 & \text{otherwise.} \end{cases}$$

Step 8. Compute additional average values for the ending part:

(11) $$m_{\text{end}}^{\text{down}} = \frac{\sum_{n=l_{\text{spl}}+1}^{N} E(n)w(n)}{\sum_{n=l_{\text{spl}}+1}^{N} w(n)},\ w(n) = \begin{cases} 1 & \text{if } E(n) < T_{\text{end}}^{\text{init}}, \\ 0 & \text{otherwise,} \end{cases}$$

(12) $$m_{\text{end}}^{\text{up}} = \frac{\sum_{n=l_{\text{spl}}+1}^{N} E(n)v(n)}{\sum_{n=l_{\text{spl}}+1}^{N} v(n)},\ v(n) = \begin{cases} 1 & \text{if } E(n) \geq T_{\text{end}}^{\text{init}}, \\ 0 & \text{otherwise.} \end{cases}$$

Step 9. Compute pair of thresholds for the beginning part:

(13) $$T_{\text{beg}}^{\text{low}} = m_{\text{beg}}^{\text{down}} + \alpha_1 (m_{\text{beg}}^{\text{up}} - m_{\text{beg}}^{\text{down}}),$$
$$T_{\text{beg}}^{\text{high}} = \max(T_{\text{beg}}^{\text{init}}, \beta_1 T_{\text{beg}}^{\text{low}}).$$

Step 10. Compute pair of thresholds for the ending part:

(14) $$T_{\text{end}}^{\text{low}} = m_{\text{end}}^{\text{down}} + \alpha_2 (m_{\text{end}}^{\text{up}} - m_{\text{end}}^{\text{down}}),$$
$$T_{\text{end}}^{\text{high}} = \max(T_{\text{end}}^{\text{init}}, \beta_2 T_{\text{end}}^{\text{low}}).$$

The parameters $\alpha_1, \beta_1, \alpha_2, \beta_2, \kappa$ and M are adjusted according to the particular requirements.

3.2. Finite state automaton

In the book about Bulgarian phonetics [33] it is claimed that no word begins with more than four consonants and also no word ends with more than three consonants. The preliminary experiments with limited amount of Bulgarian words (selected mostly from [33]) found the following: the voiced fragments can be preceded (in the beginning) and followed (in the end) by unvoiced ones with length of about 200-400 and 400-600 ms, respectively. It is worth to point out that for English language it is claimed that no word begins with more than three consonants and no current word ends with more than four consonants [30]. Also it is claimed that the mentioned above time fragments for English language are about 300 and about 500 ms, respectively [11]. The comprehensive analysis of this issue, however, is clearly outside the scope of this paper.

These two time constants are applied in the developed state automaton for defining of the pre-voiced and post-voiced fragments where the beginning and ending points will be searched. Because the values of the time constants for Bulgarian and English are close, only one pair of constants is applied no matter which speech corpus is used. The above-mentioned values of the constants are only indicative and will be adjusted during the experimental work – see Section 5.2.

The proposed ED algorithm is based on eight-state automaton with states: INIT, SCAN_DATA, SCAN_START, MAYBE_IN, SCAN_END, MAYBE_OUT, END_FOUND and END. It uses the thresholding scheme with splitting point and it is an improved version of the algorithm described by the author in [25]. A specific feature of the proposed state automaton is that in some circumstances, an error may occur. If this is happened the ED algorithm stops and the particular file is ignored in the further processing steps. The errors occur in four cases:

a) when the utterance ends outside the audio file – error ERR_TOOLONG;

b) when the SNR is very low – error ERR_LOWSPEECH;

c) when the current thresholds do not allow the starting or ending points to be found – errors ERR_BAD_BEG_THRS, ERR_BAD_END_THRS;

d) when the estimated length of the utterance is less than MinLengthTime – error ERR_TOOSHORT.

This error mechanism is designed to prevent cases when inappropriate speech data have been entered in the recognition system.

The finite state machine based decision logic applied to the ED is shown in Fig. 1. The corresponding transition rules are listed in Table. 1. The parameters $T_{\text{SCAN\_START}}$, $T_{\text{MAYBE\_IN}}$, $T1_{\text{SCAN\_END}}$, $T2_{\text{SCAN\_END}}$ and $T_{\text{MAYBE\_OUT}}$ are state timers. Each one of the time constants MaxQuietTime, UpTime1, UpTime2, MiddleTime, MinLengthTime, EndTime, BegTime, MaxStateTime determines the length of the interval after which the state transition is performed.

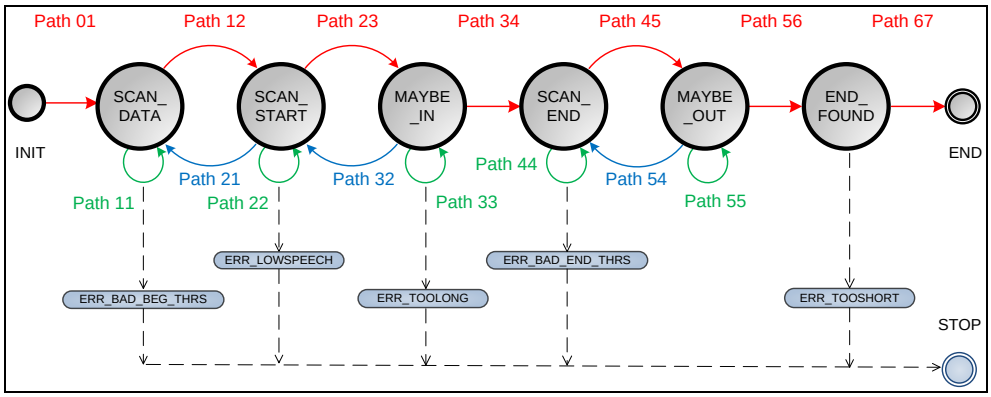

Fig. 1. Finite state machine based decision logic diagram

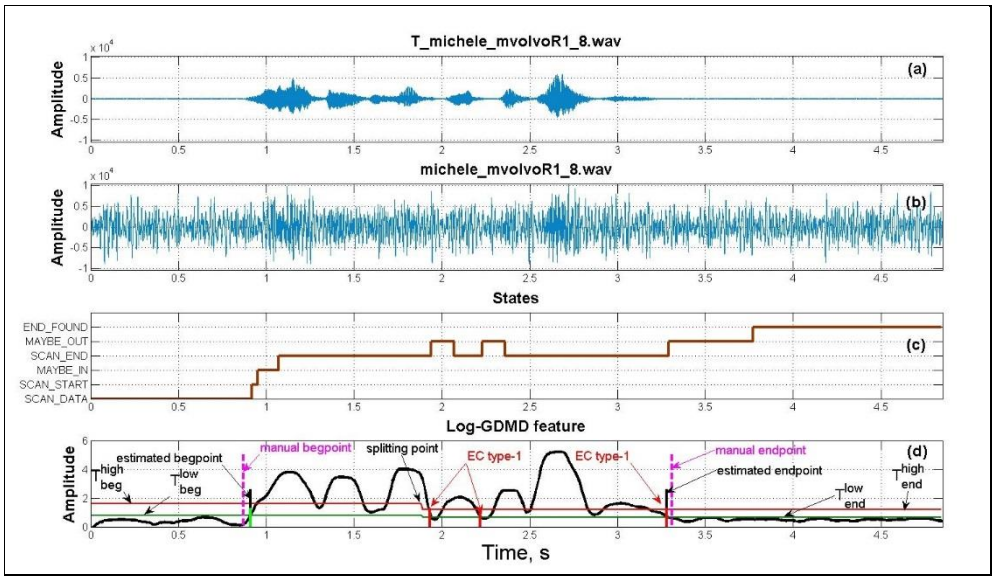

Fig. 2. Example from the SpEAR database: clean signal (a); noisy version (b); the state transition timing diagram (c); log-GDMD feature contour with marked some details in temporal execution of the algorithm (d)

In the finite state automaton algorithm two types (type-0 and type-1) of Endpoint Candidates (EC) are analyzed. The type of candidates depends on several factors. For example, the type-0 EC in the end of the utterance is typical for unvoiced sounds. The time period between the last type-1 candidate and the last type-0 candidate cannot be greater than EndTime. In Table 1 the EoF flag is set when the End-of-File has been reached.

For illustration in Fig. 2 are shown the results from the proposed ED algorithm applied on the log-GDMD feature contour for a noisy speech example selected from "Lombard Speech" section in the SpEAR database [5]. The example has clean reference and corresponded noisy version (time-aligned). It contains speech corrupted with noise, recorded inside a driving car. For the clean reference SNR = 27.00 dB and for its noisy version SNR = –14.58 dB. Both versions are downsampled to 8 kHz and are shown in Fig. 2a and b, respectively. The state transition timing diagram is shown in Fig. 2c. Along the contour in Fig. 2d are marked important details in the temporal execution of the proposed algorithm as: hand-labelled and estimated endpoints, splitting point, endpoint candidates type-1, pairs of thresholds $T_{\text{beg}}^{\text{low}}$, $T_{\text{beg}}^{\text{high}}$ for beginning part and $T_{\text{end}}^{\text{low}}$, $T_{\text{end}}^{\text{high}}$ for ending part one.

Table 1. The rules of the state transition

Paths	State Transition	Rules of State Transition	Errors
01	INIT→ SCAN_DATA	Set the beginning pair as work thresholds T_{LOW} and T_{HIGH}. Set parameters' default values and go to SCAN_DATA	
11	SCAN_DATA → SCAN_DATA	Stay in SCAN_DATA until ($E(n)<T_{LOW}$)	If current state is SCAN_DATA and EoF is set, then ERR_BAD_BEG_THRS
12	SCAN_DATA→ SCAN_START	Go to SCAN_START if ($E(n)\geq T_{LOW}$) – ***n*** is marked as beginning point candidate.	
21	SCAN_START→ SCAN_DATA	Go back to SCAN_DATA if ($E(n)<T_{LOW}$)	
22	SCAN_START→ SCAN_START	Stay in SCAN_START until ($E(n)\geq T_{LOW}$) & ($E(n)<T_{HIGH}$) & ($T_{SCAN\_START} \leq$ MaxQuietTime)	If ($E(n)\geq T_{LOW}$) & ($E(n)<T_{HIGH}$) & ($T_{SCAN_START}>$MaxQuietTime) then ERR_LOWSPEECH
23	SCAN_START→ MAYBE_IN	Go to MAYBE_IN if ($E(n)\geq T_{HIGH}$)	
32	MAYBE_IN → SCAN_START	Go back to SCAN_START if ($T_{MAYBE_IN}<$UpTime2) & ($E(n)<T_{HIGH}$)	
33	MAYBE_IN → MAYBE_IN	Stay in MAYBE_IN until ($T_{MAYBE_IN}<$UpTime2) & ($E(n)\geq T_{HIGH}$)	If current state is MAYBE_IN and EoF is set, then ERR_TOOLONG
34	MAYBE_IN → SCAN_END	if ($T_{MAYBE_IN}\geq$UpTime2) - estimate the beginning point BPoint using BegTime – then go to SCAN_END.	
44	SCAN_END → SCAN_END	Stay in SCAN_END until ($E(n)>T_{LOW}$). If $\boldsymbol{(n \geq l_{spi})}$ then set the ending pair as work thresholds T_{LOW} and T_{HIGH}.	If current state is SCAN_END, EoF is set and no EC, then ERR_BAD_END_THRS
45	SCAN_END → MAYBE_OUT	Go to MAYBE_OUT if ($E(n)\leq T_{LOW}$) – ***n*** is marked as ending point candidate	
54	MAYBE_OUT→ SCAN_END	Go back to SCAN_END if (($E(n)>T_{HIGH}$) & ($T1_{SCAN_END}\geq$UpTime1)) OR (($E(n)>T_{LOW}$) & ($T2_{SCAN_END}\geq$MiddleTime))	
55	MAYBE_OUT → MAYBE_OUT	Stay in MAYBE_OUT until (($E(n)>T_{HIGH}$) & ($T1_{SCAN_END}<$UpTime1)) OR (($E(n)>T_{LOW}$) & ($T2_{SCAN_END}<$MiddleTime))	
56	MAYBE_OUT→ END_FOUND	if ($E(n)\leq T_{LOW}$) & ($T_{MAYBE\_OUT}\geq$ MaxStateTime) - estimate ending point EPoint and utterance length ULength - then go to END_FOUND	
67	END_FOUND→ END	Go to END if (ULength≥MinLengthTime)	If (ULength<MinLengthTime) then ERR_TOOSHORT

4. Endpoint detectors

4.1. Endpoint detector GDMD-E

This detector is a combination of the log-GDMD feature and the proposed adaptive thresholds setting and state automaton. The block-diagram of the GDMD-E detector is shown in Fig. 3. The time contour of the log-GDMD feature is first computed. Thereafter the contour is analyzed and both pairs of thresholds are set. Later using the proposed state automaton and decision rules the actual utterance endpoints are estimated.

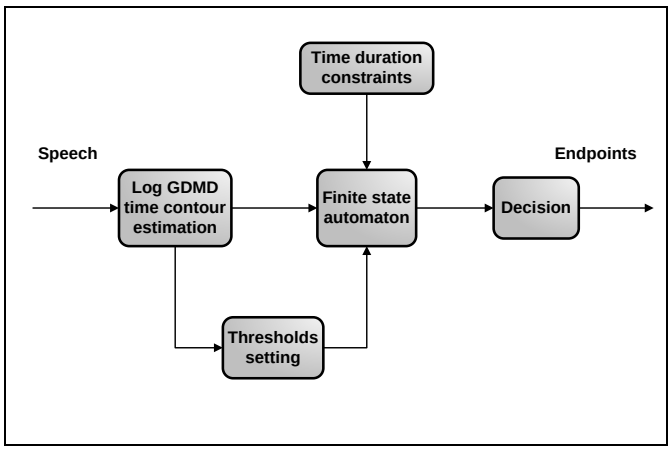

Fig. 3. Block-diagram of the log-GDMD-E endpoint detector

4.2. Endpoint detector LTSD-E

In this new detector the LTSD time contour [16] for particular file (utterance) is first computed. Then instead of using the LTSD thresholding and hangover scheme the proposed pairs of thresholds, state automaton and decision rules are applied in order to obtain the actual endpoints. The block-diagram of the LTSD-E detector is shown in Fig. 4.

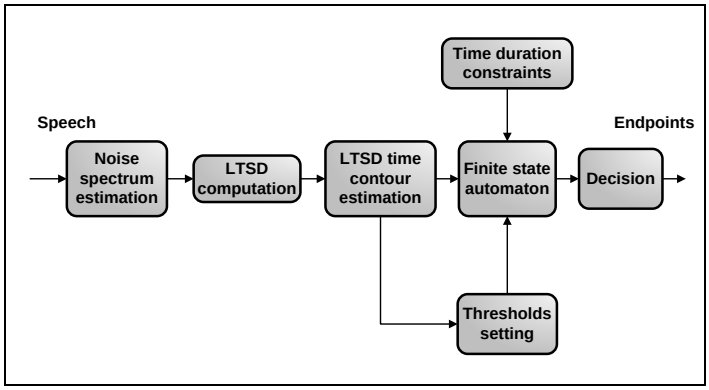

Fig. 4. Block-diagram of the LTSD-E endpoint detector

4.3. Endpoint detector GDMD-H

In the proposed detector the time contour of the log-GDMD feature is first computed. Thereafter the contour is analyzed and both pairs of thresholds are computed. The hangover technique is usually applied to smooth the speech/non-speech (1/0) decisions sequence. To obtain a similar sequence it is decided to use the two-threshold scheme with splitting point but only the high thresholds from both pairs are applied. This is done in order to perform thresholding via a single threshold. The block-diagram of the GDMD-H detector is shown in Fig. 5. In fact, the VAD is here implemented and therefore the first and last frames in the utterance labelled as speech are accepted as actual endpoints.

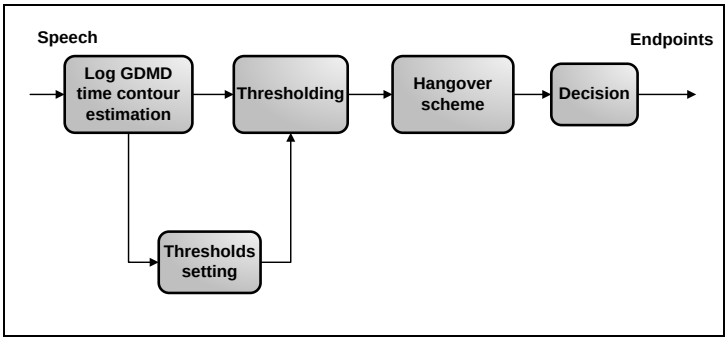

Fig. 5. Block-diagram of the GDMD-H endpoint detector

5. Experimental setup

To validate the performance of the proposed endpoint detectors two experiments were carried out. In the first one the accuracy was evaluated in terms of frame differences between hand-labelled and detected endpoints. The second experiment was conducted to evaluate the endpoint detection algorithms in terms of speaker verification performance.

5.1. Speech data

The speech data used in the experiments are selected from the BG-SRDat corpus [23] and from the TDIGITS corpus [6]. In the first experiment – accuracy evaluation – the data are chosen from both data sources, while in the second experiment – verification task – are selected only from the former one.

The BG-SRDat corpus is in Bulgarian language and it is recorded over noisy telephone channels and intended for speaker recognition. The data are sampled with frequency of 8 kHz at 16 bits, PCM format, and mono mode. The length of the utterance is about 2 s and the length of the single record (file) is about 2.5-3 s. The speech data used in the study include 337 files collected from 18 male speakers. The speech examples in the corpus do not have clean speech versions. The data are recorded in real-world environment, resulting in the fact that some records are clean and some others are very noisy.

From the TDIGITS corpus (in English) are selected examples containing spoken digit strings [6]. All examples have clean speech reference and corresponded (time-aligned) distorted versions obtained by filtering, noising and reverberation adding. The signal is sampled with frequency of 8 kHz at 16 bits, PCM format, and mono mode. The speech data used in the study include 84 files collected from 3 male and 3 female speakers. The hand-labelling of the endpoints for all speech data is done in order to have reference endpoints for comparative purposes.

5.2. Endpoint detectors parameters

The endpoint detectors parameters are tuned in the study only in the endpoint accuracy experiments (see Section 5.3). This is done in such a way that leads to a maximum rate of distribution for frame differences less than 10-frames. The tuned parameters are later used in the speaker verification tests. All adjustments are performed experimentally using a trial-and-error approach.

The signal framing is done by using the Hamming-windowed frames of 30 ms with rate of 10 ms. In FFT-spectrum calculation is chosen size of 512 points. For the log-GDMD calculation the parameters are: α= 0.6; γ = 0.4; l_w = 32 in formulas (1) and (2) and Q=3; J=6 in formulas (4) and (5), respectively.

During the experimental work some problems with the LTSD parameters have occurred. The LTSD uses calibration curve to set automatically the adaptive threshold. The parameters E_0 and E_1 are defined as the amount of noise energy in the cleanest and the noisiest conditions found in the used speech data. The values proposed in [16, 28] have led to poor performance in all tests. Similar observations were also made by the authors of the work presented in [34]. This fact made it necessary to find new sets of values that are effective for the used speech corpora. These new sets are:

1) TDIGITS: $E_0 = 70$; $E_1 = 90$; $\gamma_0 = 15$; $\gamma_1 = 10$; $\alpha = 0.95$; $\Delta_{\text{offset}} = 0$;
2) BG-SRDat: $E_0 = 60$; $E_1 = 90$; $\gamma_0 = 20$; $\gamma_1 = 6$; $\alpha = 0.95$; $\Delta_{\text{offset}} = 2$.

The initial estimation of the average noise spectrum in the LTSD calculation is done using the first 10 frames in each file assuming that they are non-speech ones.

In the hangover algorithm heuristic rules are used with timers and sequence thresholds which have the same values as in [1, 8] or B=7; S_P=3; S_L=4; L_S=5 and L_M=23, where B is the buffer length, S_P is the speech possible sequence threshold, S_L – speech likely sequence threshold, L_S –short hangover time, and L_M – medium hangover time.

The adaptive two-threshold settings algorithm uses 6 parameters, as follows: $\alpha_1 = 0.1$; $\beta_1 = 1.1$; $\alpha_2 = 0.05$; $\beta_2 = 1.2$; $\kappa = 0.5$, and M=3.

The state automaton uses 8 time constants, in ms, as follows: MaxQuietTime=2000; BegTime=300; MaxStateTime=1500; UpTime1=200; UpTime2=100; MiddleTime=200; MinLengthTime=500; EndTime=500. The constants are set in such a way that the state automaton can handle only words with length greater than MinLengthTime and phrases which include pauses with length less than MaxStateTime.

5.3. Endpoint accuracy experiments

In this experiment the endpoints accuracy was evaluated in terms of frames differences between hand-labelled and detected endpoints [37]. The frames differences are denoted as D_B and D_E – for beginning and ending points, respectively. The histograms of D_B and D_E for the two corpora are presented in Figs. 6-7 and Figs 8-9, respectively. In Tables 2 and 3 is shown the statistical information of the histograms – each value shows the rate of distribution in percentages for all test conditions. The absolute values of the differences are denoted in the Tables 2 and 3 as $|D_B|$ and $|D_E|$ while with $\overline{D}$ are denoted their average values for the particular feature and the corresponding frame difference. Two labels – "skip" and "add"– are added to the histograms. They are used to denote the areas in histograms corresponding to the skipped or to the added frames. The data points from each corpus used for the histograms' creation are 84 and 262 and the final number of bins are 9 and 19, respectively. These numbers are the averages of the number of bins calculated for each feature by the Scott's normal reference rule [31].

The selected utterances from the TDIGITS corpus containing different digit strings so they start and end with different phonemes. For this corpus according to Table 2 the maximum average rate $\overline{D}$ for both differences belongs to the GDMD-E detector. The maximum value in each column is in bold text in Tables 2 and 3.

The phrase from the BG-SRDat corpus starts with voiced fricative "z" and ends with unvoiced fricative "*s*" [23]. As seen in Figs. 8 and 9 there are substantial peaks (for almost all detectors) placed in the so called "add-area" in each histogram where the non-speech frames are added. According to Table 3 the maximum average rate $\overline{D}$ for both differences belongs to the GDMD-E detector.

As seen in Figs. 6-9 all detectors add non-speech segments at the both ends of the utterances. Most non-speech segments are added at the end of the utterances by the LTSD-H detector, while at least are added by the GDMD-E one.

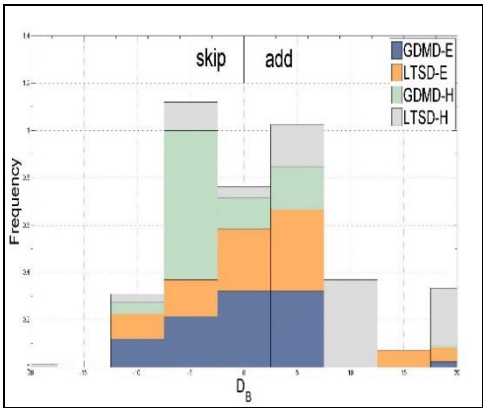

Fig. 6. The histograms of D_B-TDIGITS corpus

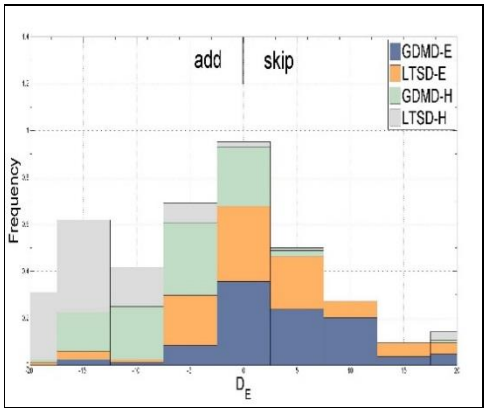

Fig. 7. The histograms of D_E-TDIGITS corpus

In other words, if the phrase starts or ends with fricative consonants, the endpoint detectors that use the hangover scheme prolong the utterances more than the detectors using the proposed state automaton algorithm. This fact is due to the one significant difference in the mode of operation of these two algorithms. In the state automaton algorithm are formed sets of reliable thresholds' crossing points. The actual endpoints are selected from these sets based on the rules, i.e., the endpoints are always selected from the real crossing points. This is the main difference compared to the hangover scheme where the speech fragments initially obtained by thresholding are extended by heuristic rules.

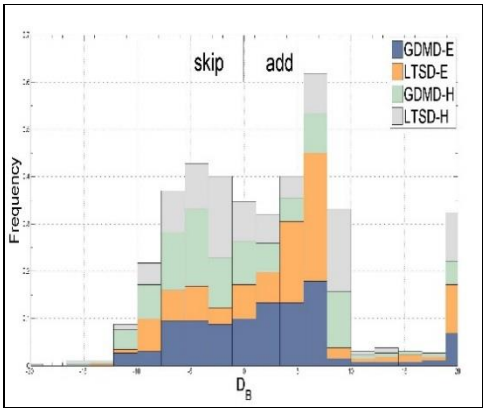

Fig. 8. The histograms of D_B-BG-SRDat corpus

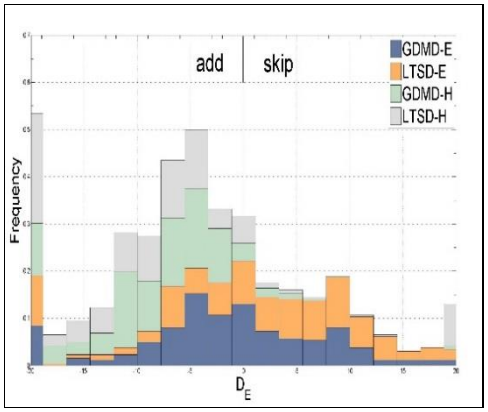

Fig. 9. The histograms of D_E-BG-SRDat corpus

Table 2. The rate of distribution in percentages – TDIGITS

Corpus	TDIGITS					
Endpoint detector	$\|D_B\|$		$\|D_E\|$		$\bar{D}$	
	≤ 5	≤ 10	≤ 5	≤10	≤ 5	≤ 10
GDMD-E	85.71	97.61	67.85	**89.28**	**76.78**	**93.45**
LTSD-E	76.19	86.90	**76.19**	84.52	76.19	85.71
GDMD-H	**94.04**	**98.80**	58.33	80.95	76.19	89.88
LTSD-H	34.52	75.00	11.90	28.57	23.21	51.78

Table 3. The rate of distribution in percentages – BG-SRDat

Corpus	BG-SRDat					
Endpoint detector	$\|D_B\|$		$\|D_E\|$		$\bar{D}$	
	≤ 5	≤ 10	≤ 5	≤10	≤ 5	≤ 10
GDMD-E	**54.96**	**87.02**	**51.90**	**78.24**	**53.43**	**82.63**
LTSD-E	41.60	84.35	37.02	67.17	39.31	75.76
GDMD-H	47.32	**87.02**	35.11	61.06	41.22	74.04
LTSD-H	45.80	85.11	24.42	46.56	35.11	65.83

5.4. Speaker verification experiments

The proposed endpoint detectors are examined with the help of two fixed-phrase speaker verification applications [32]. Their brief general descriptions are given in the text below.

5.4.1. Pre-processing step (for both applications)

In the pre-processing step the Hamming-windowed frames of 30 ms with rate of 10 ms are used. The number of the Mel-Frequency Cepstral Coefficients (MFCC) is 14 (without zeroth coefficient) and the cepstral mean subtraction is applied for each file separately [9].

5.4.2. DTW speaker verification

This application is based on the normalize-wrap DTW algorithm with the root power sum – cepstral distance [21]. The speaker's reference is obtained by an averaging (after dynamic time warping alignment) of his training utterances [40]. The speakers' verification thresholds are estimated by using of the cohorts with size of 3 [4].

5.4.3. HMM speaker verification

The phrase modeling in this application is done by a whole-phrase continuous HMM [2, 10, 27]. The selected model is with a left-to-right topology with no skip state and the output distributions are represented as mixture of Gaussians with diagonal covariance matrices. The HMM training is carried out by well-known Baum-Welch Algorithm [10, 27].

In the verification are used the individual speaker's thresholds. They are estimated by using of the world (or background) model as non-speaker model. The speaker's score is obtained by computing the log-likelihood ratio of the particular utterance using the speaker and world models. The verification thresholds are set a priori based on the scores distributions from claimed speakers and impostors [22, 26].

5.4.4. Speech data

The speech data used in the speech verification experiments include 337 records of the phrase collected from 18 male speakers. The bigger part of them – 262 records from 12 speakers (these data are the same for both applications) is intended for templates or models forming (training set), for thresholds settings (validation set) and for testing (verification set). Because the speech corpus is small the same data set is used for training and for validation [3]. The rest of data – 75 records from 6 speakers are selected for the world model training in the HMM test.

The 5×2 fold cross validation method is applied in order to make an efficient use of all available data [7]. Overall results are computed as weighted means of the outcomes from the five repetitions. In the verification mode there are 142 client accesses or false rejection tests and 1562 impostor accesses or false acceptance tests. After five runs the total tests are: for false rejection – 710, and for false acceptance – 7810.

5.5. Experimental results

The performance of various endpoints detectors is compared via the verification results in the study. Additional verification task is done with hand-labelled endpoints.

For limited real-world data the single value error is not reliable estimation of the speaker verification performance [3]. Since this is our case, it was decided to apply the methodology for performance estimation of the speaker verification proposed in

[3]. The verification results are presented as rate ratios – False Rejection Rate (FRR), False Acceptance Rate (FAR) and the Half Total Error Rate (HTER). Also the 95% Confidence Interval (CI) for the HTER is shown computed according to [3]. The Z_{HTER} -test method proposed in [3] is applied to verify whether the given classifier is statistically significantly different than another.

In this test, the performance of various endpoints detectors is compared via the verification rate, i.e., for each endpoint detector a separate speaker verification task is carried out. When comparing the verification rates obtained from systems that have many identical modules and training and testing data are the same, must be considered that the obtained scores should be strongly correlated [3]. Given this and according to the comments in [3] the independent case of the Z_{HTER} -test is applied – mainly because its results are not optimistically biased.

In the text below, for the sake of clarity, some sentences are simplified, e.g., "detector GDMD-E provides better verification results" means that the speaker verification application that uses this detector, provides better verification results, not the detector itself.

The DTW speaker verification results are shown, as follows: in Table 4 – the rates and the confidence intervals for the HTERs; in Table 5 – the confidence values in percentages for each pair of detectors.

Table 4. DTW speaker verification results

Endpoint detector	FRR, %	FAR, %	HTER, %	95%CI
Manual	7.88	4.83	6.35	±0.0101
GDMD-E	9.43	8.22	8.82	±0.0111
LTSD-E	10.70	11.57	11.13	±0.0119
GDMD-H	10.84	6.42	8.63	±0.0117
LTSD-H	11.83	11.65	11.74	±0.0124

Table 5. Confidence values in percentages for each pair of detectors – DTW test

Endpoint Detectors	Manual	GDMD-E	LTSD-E	GDMD-H	LTSD-H
Manual	–	99.86	100.00	99.58	100.00
GDMD-E	–	–	99.41	18.63	99.93
LTSD-E	–	–	–	99.66	50.96
GDMD-H	–	–	–	–	99.96
LTSD-H	–	–	–	–	–

It is known that the number of states in the HMM is chosen empirically and it is good practice to be proportional to the number of phonemes in the pass-phrase [2]. The used Bulgarian phrase includes 6 different words comprising a total of 31 phonemes – 10 vowels and 21 consonants. The phrase includes also five pauses between the words. In order to define the HMM topology are carried out preliminary experiments with different number of states and mixtures –18, 25, 35 and 2, 3, 4, respectively. The experiments are done using the hand-labelled utterances. The minimal verification error (HTER = 8.42%) is obtained for the left-to-right HMM with 35 states and two Gaussian mixtures and this topology is used in all experiments.

The HMM speaker verification results are shown, as follows: in Table 6 – the rates and the confidence intervals for the HTERs; in Table 7 – the confidence values in percentages for each pair of detectors.

Table 6. HMM speaker verification results

Endpoint detector	FRR, %	FAR, %	HTER, %	95%CI
Manual	15.63	1.21	8.42	±0.0134
GDMD-E	18.45	0.98	9.71	±0.0143
LTSD-E	22.25	1.20	11.72	±0.0153
GDMD-H	18.45	1.02	9.73	±0.0143
LTSD-H	22.53	1.04	11.78	±0.0154

Table 7. Confidence values in percentages for each pair of detectors – HMM test

Endpoint Detectors	Manual	GDMD-E	LTSD-E	GDMD-H	LTSD-H
Manual	–	80.44	99.85	81.12	99.87
GDMD-E	–	–	93.95	1.54	94.63
LTSD-E	–	–	–	93.69	4.31
GDMD-H	–	–	–	–	94.39
LTSD-H	–	–	–	–	–

The averaged DET curves [19] are plotted in Fig. 10 and Fig. 11 to analyze the DTW and HMM verification performance obtained for each endpoint detector.

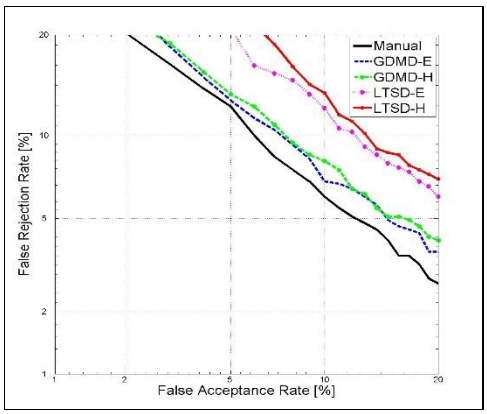

Fig. 10. DET curves for different EDs – DTW tests

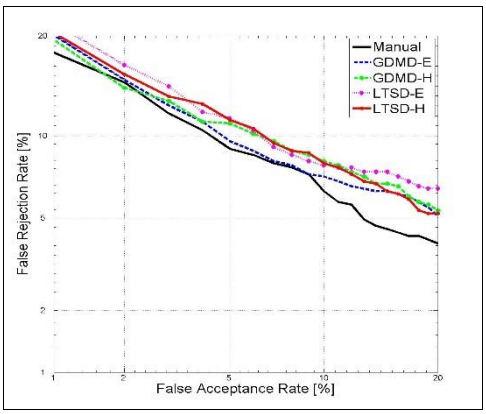

Fig. 11. DET curves for different EDs – HMM tests

5.6. Comments

Figs 10 and 11 clearly show that the both GDMD-based detectors curves are closer to the reference curve (manual ED) than the other two. These results are consistent with those found in the Tables 4 and 6, i.e., the GDMD-E and the GDMD-H detectors performs the best in both speaker verification tests. Of the two detectors the GDMD-E is better in the endpoint accuracy test for both corpora (Tables 1, 2) and HMM test (Table 6), while the second one is slightly better in the DTW verification test (Table 4). Based on Z_{HTER} -tests both detectors are equivalent. The worse verification rates of the LTSD-E and LTSD-H detectors are caused by the serious endpoint

detection errors. As seen in Figs 6-9 they demonstrate a lot of frames differences that are greater than 20 frames (200 ms).

In the study the following observations are made from the results obtained:

- in all tests, the log-GDMD-based detectors demonstrated always better performance than the LTSD-based ones;
- in accuracy tests, the best results (maximum average rates) are obtained by the GDMD-E detector for both corpora;
- the best rate in the speaker verification experiments is obtained by the GDMD-H detector in the DTW tests, and by the GDMD-E detector in the HMM tests;
- the Z_{HTER} -tests showed that the GDMD-based detectors provide results that are always statistically significantly different from the LTSD-based ones. The pairs of detectors: [GDMD-E, LTSD-E]; [GDMD-E, LTSD-H]; [GDMD-H, LTSD-E]; [GDMD-H, LTSD-H] gave in the DTW and HMM trials the confidence values greater than 90%.
- the Z_{HTER} -tests showed that among the analyzed detectors, more important part (in statistical sense) of them is the feature, not the decision scheme. The pairs of detectors with the same features, but with different decision algorithms always are equivalent. That is true for pairs [GDMD-E, GDMD-H] and [LTSD-H, LTSD-E] that provided very low confidential values.

6. Conclusions

In the study the effectiveness of four contour-based single-feature speech endpoint detection algorithms was evaluated experimentally. One of them is the well-known LTSD algorithm with hangover scheme. The other three are new and are designed using the combination of different features (the log-GDMD and the LTSD) and different decision schemes (the hangover algorithm and the eight-state automaton with adaptive two-threshold scheme).

The experiments revealed three facts:

- First, the log-GDMD-based endpoint detectors always (in all tests) perform better than the LTSD-based ones. It is worth to note that the LTSD feature is adaptive to the varying noise levels while the log-GDMD one relies only on the intrinsic noise robustness of the two its components – the modified group delay spectrum and the delta spectral autocorrelation function.
- Second, in the endpoint detection accuracy tests the state automaton with adaptive threshold scheme outperforms the hangover scheme for the same features.
- Third, in speaker verification tests for the same features, the state automaton with adaptive threshold scheme mostly outperforms the hangover scheme in terms of verification rate but difference between them is not statistically significant.

Future work will be focused on two directions: (1) development of the endpoint detector which integrates the GDMD feature and the pitch harmonics-based features, and (2) study of the GDMD-H detector as a part of the VAD algorithm in the neural network-based text-independent speaker recognition framework.

References

1. B a g i n s k i, M. Robust Speech Detection in High Levels of Background Noise. Imperial College, Department of Electrical and Electronic Engineering, Final Year Project Report, 2014, pp. 1-67.
2. B e u y e u k, O., M. A r s l a n. Model Selection and Score Normalization for Text-Dependent Single Utterance Speaker Verification. – Turkish Journal of Electrical Engineering & Computer Sciences, Vol. **20**, 2012, No Sup. 2, pp. 1277-1295.
3. B e n g i o, S., J. M a r i e t h o z. A Statistical Significance Test for Person Authentication. – In: Proc. of ODYSSEY – The Speaker and Language Recognition Workshop, 2004, pp. 237-244.
4. B u r i l e a n u, C., et al. On Performance Improvement of a Speaker Verification System Using Vector Quantization, Cohorts and Hybrid Cohort-World Models. – International Journal of Speech Technology, Vol. **5**, 2002, pp. 247-257.
5. Center for Spoken Language Understanding, Speech Enhancement and Assessment Resource (SpEAR) Database. Oregon Graduate Institute of Science and Technology, September 2016. **http://www.cslu.ogi.edu/nsel/data/SpEAR_lombard.html**
6. Dan Ellis's Home Page, Sound Examples for Projects. Columbia University, August 2017. **https://www.ee.columbia.edu/~dpwe/**
7. D i e t t e r i c h, T. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. – Neural Computation, Vol. **10**, 1998, No 7, pp. 1895-1923.
8. ETSI, Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithms. ETSI ES 202 050 V1.1.5 (2007-01). Annex A.3. Stage 2 – VAD Logic. 2007, pp. 42-43.
9. G a n c h e v, T. Contemporary Methods for Speech Parameterization. – Springer Briefs in Speech Technology. New York, Springer-Verlag, 2011. 114 p.
10. G a l e s, M., S. Y o u n g. The Application of Hidden Markov Models in Speech Recognition. – Journal Foundations and Trends in Signal Processing, Vol. **1**, 2008, No 3, pp. 195-304.
11. G h a e m m a g h a m i, H., et al. Noise Robust Voice Activity Detection Using Normal Probability Testing and Time-Domain Histogram Analysis. – In: Proc. of IEEE ICASSP, 2010, pp. 4470-4473.
12. H e g d e, R., H. M u r t h y, V. G a d d e. Significance of the Modified Group Delay Feature in Speech Recognition. – IEEE Transactions on ASLP, Vol. **15**, 2007, No 1, pp. 190-202.
13. J i a, C., B. X u. An Improved Entropy Based Endpoint Detection Algorithm. – In: Proc. of ISCSLP, 2002, pp. 96-100.
14. K i t a o k a, N., et al. Development of VAD Evaluation Framework CENSREC-1-C and Investigation of Relationship between VAD and Speech Recognition Performance. – In: Proc. of IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU'07), 2007, pp. 607-612.
15. K y r i a k i d e s, A., et al. Isolated Word Endpoint Detection Using Time-Frequency Variance Kernels. – In: Proc. of Conference Record of the 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR'11), 2011, pp. 1041-1045.
16. L u e n g o, I., et al. Modified LTSE-VAD Algorithm for Applications Requiring Reduced Silence Frame Misclassification. – In: Proc. of International Conference on Language Resources and Evaluation (LREC'10), 2010, pp. 1539-1544.
17. L i, J., Z. P i n g, J. X i n x i n g, D. Z h i r a n. Speech Endpoint Detection Method Based on TEO in Noisy Environment. – Procedia Engineering, Vol. **29**, 2012, pp. 2655-2660.
18. L i, Q., J. Z h e n g, A. T s a i, Q. Z h o u. Robust Endpoint Detection and Energy Normalization for Real-Time Speech and Speaker Recognition. – IEEE Transaction on SAP, Vol. **10**, 2002, No 3, pp. 146-157.
19. M a r t i n, A., et al. The DET Curve in Assessment of Detection Task Performance. – In: Proc. of EUROSPEECH, 1997, pp. 1895-1898.
20. M e s a-N a v a r r o, J., et al. An Improved Speech Endpoint Detection System in Noisy Environments by Means of Third-Order Spectra. – IEEE SP Letters, Vol. **6**, 1999, No 9, pp. 224-226.

21. M y e r s, C., L. R a b i n e r, A. R o s e n b e r g. Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition. – IEEE Transactions on ASSP, Vol. **28**, 1980, No 6, pp. 623-635.
22. M u n t e a n u, D., S. T o m a, Automatic Speaker Verification Experiments Using HMM. – In: Proc. of 8th International Conference on Communications, 2010, pp. 107-110.
23. O u z o u n o v, A. BG-SRDat: A Corpus in Bulgarian Language for Speaker Recognition over Telephone Channels. – Cybernetics and Information Technologies, Vol. **3**, 2003, No 2, pp. 101-108.
24. O u z o u n o v, A. Noisy Speech Endpoint Detection Using Robust Feature. – In: Proc. of Springer International Publishing Switzerland. V. Cantoni et al., Eds. BIOMET 2014, LNCS 8897, 2014, pp. 105-117.
25. O u z o u n o v, A. Telephone Speech Endpoint Detection Using Mean-Delta Feature. – Cybernetics and Information Technologies, Vol. **14**, 2014, No 2, pp. 127-139.
26. P a r t h a s a r a t h y, S., A. R o s e n b e r g. General Phrase Speaker Verification Using Sub-Word Background Models and Likelihood-Ratio Scoring. – In: Proc. of ICSLP'96, pp. 2403-2406.
27. R a b i n e r, L. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. – Proceedings of the IEEE, Vol. **77**, 1989, No 2, 257-286.
28. R a m i r e z, J., et al. Efficient Voice Activity Detection Algorithms Using Long-Term Speech Information. – Speech Communication, Vol. **42**, 2004, No 3-4, pp. 271-287.
29. R a m i r e z, J., et al. SVM-Based Speech Endpoint Detection Using Contextual Speech Features. – Electronics Letters, Vol. **42**, 2006, No 7, pp. 426-428.
30. R o a c h, P. English Phonetics and Phonology: A Practical Course. 4th Ed. Cambridge University Press, 2009. 283 p.
31. S c o t t, D. Scott's Rule. – WIREs Computational Statistics, Vol. **2**, 2010, pp. 497-502.
32. Speaker Recognition Project – Proprietary Developed Software. Internal Reports (unpublished), Fadata, Ltd., 2003.
33. T i l k o v, D., T. B o j a d z h i e v. Balgarska Fonetika (Bulgarian Phonetics), Sofia, Nauka i Izkustvo, 1977 (in Bulgarian).
34. T u o n o n e n, M., R. H a u t a m ä k i, P. F r ä n t i. Automatic Voice Activity Detection in Different Speech Applications. – e-Forensic, Vol. **12**, 2008, pp. 1-6.
35. W u, G., et al., Fuzzy Neural Networks for Speech Endpoint Detection. – In: Proc. of 2012 International Conference on Fuzzy Theory and Its Applications, 2012, pp. 354-356.
36. W u, B., K. W a n g. Robust Endpoint Detection Algorithm based on the Adaptive Band-Partitioning Spectral Entropy in Adverse Environments. – IEEE Transactions on SAP, Vol. **13**, 2005, No 5, pp. 762-775.
37. Y a m a m o t o, K. et al. Robust Endpoint Detection for Speech Recognition Based on Discriminative Feature Extraction. – In: IEEE ICASSP, Vol. **I**, 2006, pp. 805-808.
38. Y a l i, C. et al. A Speech Endpoint Detection Algorithm Based on Wavelet Transforms. – In: Proc. of 26th Chinese Control and Decision Conference (CCDC'14), 2014, pp. 3010-3012.
39. Z h a n g, T., H. H u a n g, L. H e, M. L e c h. A Robust Speech Endpoint Detection Algorithm Based on Wavelet Packet and Energy Entropy. – In: Proc. of 3rd International Conference on Computer Science and Network Technology, 2013, pp. 1050-1054.
40. Z e l i n s k i, R., F. C l a s s. A Learning Procedure for Speaker-Dependent Word Recognition System Based on Sequential Processing of Input Tokens. – In: Proc. of IEEE ICASSP, 1983, pp. 1053-1056.
41. Z h a n g, Z., S. F u r u i. Noisy Speech Recognition Based on Robust End-Point Detection and Model Adaptation. – Proceedings of IEEE ICASSP, Vol. **1**, 2005, pp. 441-444.
42. Z h u, J., F. C h e n. The Analysis and Application of a New Endpoint Detection Method Based on Distance of Autocorrelated Similarity. – In: Proc. of EUROSPEECH, 1999, pp. 105-108.