Project Thesis

Automatic Segmentation of Speech into Sentences Using Prosodic Features

Florian Pausch

submitted to the
Institute of Electronic Music and Acoustics
of the
University of Music and Performing Arts Graz

in November 2011

Supervisor:
DI Johannes Luig

kunst uni graz

institute of electronic music and acoustics iem

Abstract

Segmentation of speech into sentences plays an important role as a first step in several speech processing fields. Automatic Speech Recognition (ASR) algorithms mostly produce just a stream of non-structured words without detecting the hidden structure in spoken language. However, natural language processing devices often have a strong need for sentence-like units to work properly. Apart from that, hand-labeling is very time-consuming. Thus, it is reasonable to develop an algorithm which marks sentence and phrase boundaries using prosodic features. In this project thesis, the Aix-MARSEC database of BBC radio speech is used for analysis.

The algorithm can be described as following: An adaptive, energy-based voice-activity-detector (VAD) is used to gather all active regions and calculate the pause lengths and intensity as first features. These blocks are then used as input for a pitch estimation algorithm. To assess tendencies at the region boundaries, we calculate an optimal (in the least-squares sense) piecewise polynomial approximation and derive various prosodic features (initial/final intonation, pitch gradient, downdrift...). Consequently, the extracted features are combined in a decision tree to determine the sentence boundaries.

Zusammenfassung

Automatische Satzsegmentierung von Sprache stellt einen wichtigen ersten Schritt in vielen Bereichen der Sprachsignalverarbeitung dar. Spracherkennungsprogramme geben meist nur die Grenzen von Wörtern aus ohne vorhandene Strukturen wie Satzgrenzen zu detektieren. In der linguistischen Sprachverarbeitung werden jedoch genau solche Grenzen benötigt, damit Programme zuverlässig funktionieren. Abgesehen davon ist es sehr zeitaufwendig, Satzgrenzen per Hand zu labeln. Ziel ist deshalb die Entwicklung eines Algorithmus, der eine Satzsegmentierung unter Verwendung von prosodischen Merkmalen durchführt. In dieser Projektarbeit wird dabei die Aix-MARSEC Datenbank, eine Sammlung von BBC Nachrichten, einer Analyse unterzogen.

Der Algorithmus gliedert sich dabei folgendermaßen: Stimmhafte Regionen werden zunächst mittels eines adaptiven, energiebasierenden Spracherkennungsalgorithmus detektiert und die Pausenlängen bzw. die Intensität als erste Features berechnet. Die gefundenen Blöcke stellen die Regionen für die nachfolgende Tonhöhenanalyse dar. An den Grenzen werden Tendenzen im Tonhöhenverlauf mittels linearer Regression (optimal im Sinne eines kleinsten quadratischen Fehlers) berechnet und daraus verschiedene prosodische Merkmale (Intonation an Satzgrenzen, Gradient der Intonation, Grundtonabfall...) abgeleitet werden. In weiterer Folge werden diese Merkmale in einem Entscheidungsbaum kombiniert und die Satzgrenzen ermittelt.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Aims

Segmentation of speech into sentence-like units is used in many areas of speech signal processing. Purely applied to spoken language, sentence segmentation is quite complex because of absent punctuation marks and the lack of other additional typographic indicators. Recent works showed that hidden structures can be found more efficiently when using prosody-based features [SSHTT00].
Text segmentation is needed for several algorithms in natural language processing and thus incorporates a first crucial step for thorough speech analysis like topic segmentation [XYL+10], morphological analyis [Jua10], parsing [HPW04] or various information retrieval algorithms [GFCP07].
Based on this micro division, the next logical step is the coalescence of found tokens to bigger coherent units such as sentences. Automatic sentence segmentation based on prosody is not a trivial task, as the speaking style is language dependent (socio-cultural aspects, idioms, prosody...) [WLL09, KFV+03] and strongly constrained by speech type (broadcast speech, spontaneous speech...) [PL97, Lli92] – not considering other challenges (background noise, speaker turns...) [WL01, SSRS01, DC11] at this point.
However, given the data amounts of speech corpora (in this case the Aix-MARSEC database is analyzed), it is obvious that hand labeling consumes way too much time, so the implementation of an automatic sentence segmentation algorithm suggests itself.
Another future challenge can be found in the real-time extraction of prosodic features (as they are defined within sentence-like units) to analyze emotion, workload or stress of a speaker.

1.2 Structure of the Project Thesis

After a short introduction to prosodic features (section 2), the most important parts of the algorithm are summarized (section 3). To go into detail, section 3.1 covers the first major functional block, an adaptive energy-based voice activity detection algorithm to gather voiced and unvoiced regions out of a continuous speech stream. The extraction of prosodic features heavily relies on a fundamental frequency estimator which is described in section 3.2. By using a modified autocorrelation function which is integrated in the so-called YIN-algorithm, a pitch track is determined. Hence, with the help of linear regression analysis (appendix A), the prosodic features are extracted. All derived features are then combined and processed within a decision tree (section 4) which outputs assessed sentence boundaries. The final chapter (section 5) evaluates the algorithm by comparing the results with hand-labelled boundaries.

1.3 The Aix-MARSEC Database

The "Aix-MARSEC" project [ABH04] comprises a collection of spoken British English which makes up a freely available database. Altogether it consists of five and a half hours of BBC news material.
It is well suited for this project thesis as it is fully annotated (including sentence boundaries) and labeled on different linguistic levels (phonemes, syllables, words, rhythmic units, intonation coding...). Additionally, eleven different speaking styles by 17 female and 36 male speakers are useful to evaluate the implemented algorithm under various conditions. Apart from that, the database has an evolutionary character meaning that many users contributed to and expanded the open-source project (GNU General Public License).

1.4 Programming Environment

The realization of this project is carried out offline by use of the software MATLAB™ (R2010b) by *The MathWorks®*. It not only offers a wide range of powerful mathematical manipulation possibilities (optimized for vector and matrix computations) but also enables graphical output to observe the algorithm's functionality and parameter changes.

2 Prosodic Features

Prosodic cues represent an important factor within this work, as they can be consulted as an indicator for sentence boundaries [SSHTT00].
In linguistics, the term *prosody* refers to suprasegmental phenomena in speech, meaning that one has to look at entire utterances rather than just considering phonemes, syllables or words. From the acoustic viewpoint, *prosody* means the alteration of syllable length, pitch, rhythm or loudness [CHJC+06].
Speakers additionally use prosody to impress some sort of emotion and attitude (e.g. sarcasm, irony...) which is not obviously integrated in the text – in short, the focus lies on "the way we say something" and not on "what we say". Prosody can also be consulted to classify a sentence as a question, a command or just a statement by analyzing the interior pitch track progression. Stressing of certain words or passages leads to a contextual emphasis of what is said, whereas inter-word pauses or other temporal properties often are hints for syntactic structures.

2.1 Extraction Points

The feature extraction points are determined by inter-word pauses (derived from the VAD-algorithm, section 3.1) with a length of more than $t_{anal} = 100$ ms[1]. This value has been determined experimentally and was chosen in order to include any potential sentence boundary during analysis process. The regions preceding and following the pause are then analyzed by means of linear approximation (appendix A) to retrieve several pitch features.

In the following, the energy-based (section 2.2) and pitch-related (section 2.3) prosodic features which are used in this project are described.

2.2 Energy Features

2.2.1 Short-term Intensity

Many perceived prosodic features correlate with measurable signal properties. For short-term intensity, short-term energy is the accordant attribute. There are mainly three mechanisms that control energy in speech production: (1) variation of lung pressure, (2) larynx adjustment and (3) vocal tract adaption [LD98]. As a measure for the short-term signal energy, we calculate the root mean squared (RMS) value for blocks of N samples[2] (equation 3.1, page 11).
As a result, we get a time varying intensity track (figure 2.1, upper plot) which is important for

[1] This parameter is stored in a MATLAB structure array called *Advanced*.
[2] The value for N is commonly chosen such that $N/f_s = 20$ ms. This value should guarantee that the short-term energy within one signal period ($N/f_s < 10$ ms) is not tracked, but it is as well short enough to extract the quasi-periodic part of a speech signal ($N/f_s < 30$ ms) [PR09].

the voice activity detection algorithm (section 3.1). The representation in [dB] approximates the perception of the human ear.

2.2.2 Energy Decay

Usually, speakers tend to finish long sentences with less energy, rather than taking an additional breath towards the end of a sentence. This usually leads to a measurable decay of short-term energy within voiced regions and is calculated by means of linear approximation (figure 2.1, appendix A).

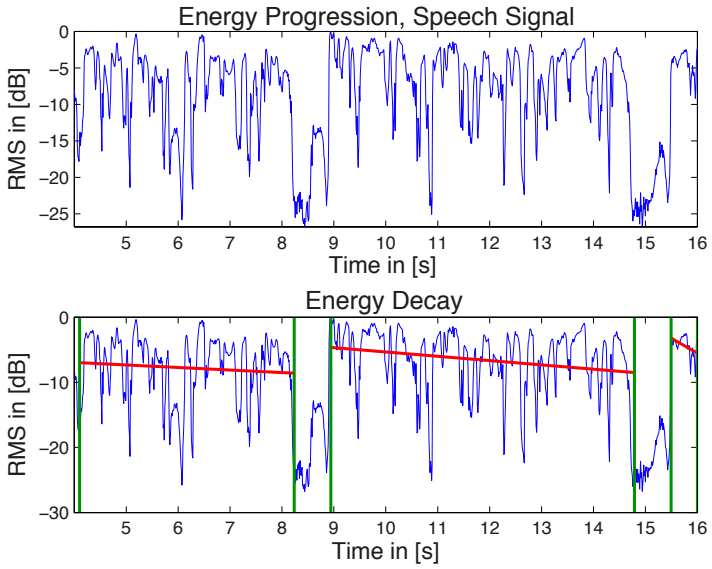

Figure 2.1: Above: frame-based energy progression of a speech signal [dB], Below: energy decay (red) in [dB/s] within potential sentence boundaries (green)

For proper usage within the decision tree (section 4), the feature is normalized in the following way:

$$EnergyDecay_{d,norm} = \frac{|EnergyDecay_d|}{\frac{1}{D}\sum_{d=1}^{D} EnergyDecay_d}, \tag{2.1}$$

with D comprising the number of (potential) sentences. A remarkable variation of the energy decay between two potential sentence boundaries can be well observed and compared by looking at the normalized feature, given in [dB/s]. Thus, it can be used as an indicator for sentence boundaries.

2.2.3 Pause Length

Pauses are defined by voiceless regions within a continuous speech stream which additionally fall below an adaptively found threshold E_{thr} (section 3.1) with respect to the RMS level (figure 2.2). The longer voiceless regions are, the more probably a sentence boundary is found. This feature is very important on the one hand to define extraction points for further prosodic feature calculation and on the other hand (in case of a conspicuously long pause[3]) to detect a sentence boundary.

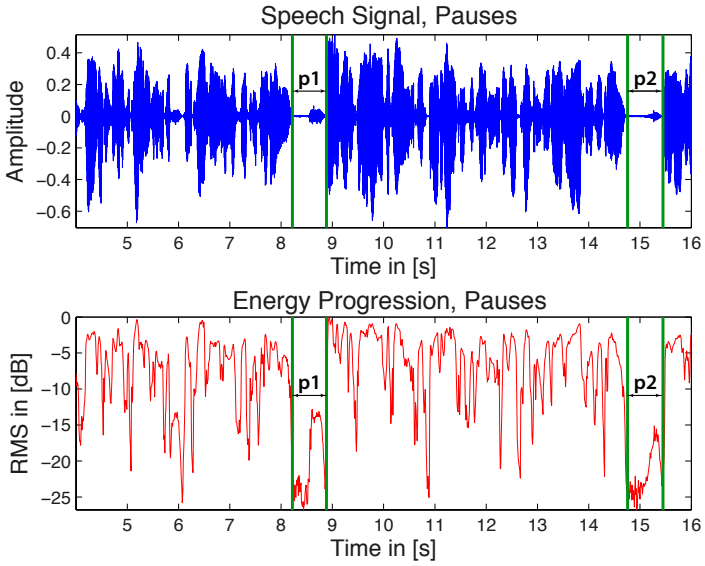

Figure 2.2: Pause lengths p_1, p_2 (defined by voiceless regions within a speech signal; inter-word pauses are not marked)

The normalization of this feature is not a trivial task, as pause lengths vary subject to individual speaking styles and therefore needs to be analyzed independently for each speaker.

2.3 Pitch Features

Pitch is a perceptual parameter that makes it possible to arrange sounds on a frequency-related scale. In physical terms, the melody of speech originates from a measurable pitch track made up by the fundamental frequency f_0. This definition only exists for periodic or nearly periodic signals and can be obtained by inverting the signal's period [KD06].
In the algorithm, the voiced parts of the input signal excerpt are extracted by applying the VAD algorithm (section 3.1) and analyzed with respect to *pitch* by a fundamental frequency estimator (section 3.2).

[3] The user is able to tune a parameter called *MinPause* [s]. If a pause exceeds this length the involved boundaries will be classified as sentence boundaries in the decision tree (section 4).

For prosodic analysis, it is important to check the intonation before and after the detected extraction points (boundary intonation). Hence, a trend line fitting is applied on the last part of the preceding analysis block and the first part of the consecutive one (250 ms, respectively[4]).

2.3.1 Initial and Final Intonation

A first pitch-related feature is the intonation at the beginning of a sentence (*initial rise / fall*) and towards the end (*final lowering / rise*), respectively; measured in [Hz] (figure 2.3). In broadcast corpora, we are mostly dealing with statements without questions, so it is expected that most sentences end with a final lowering. The melody at the beginning of a sentence strongly depends on speaking style and does not necessarily stick to general rules.
Subsequent gradient calculation (in [Hz/s]) shows the tendency and the variation amount of the observed boundary intonation and can be consulted as an indicator for sentence boundary validation.

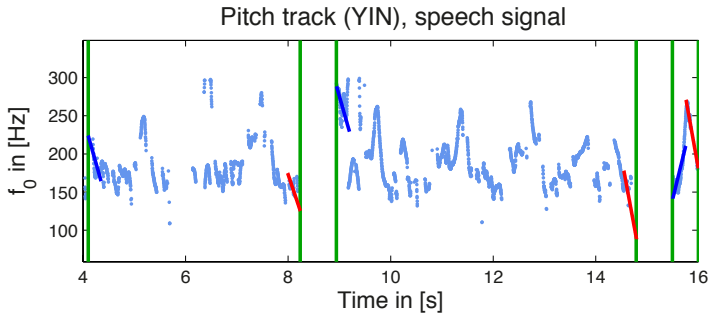

Figure 2.3: Intonation at potential sentence boundaries;
light blue: pitch track, blue: initial intonation,
red: final intonation, green: potential sentence boundaries

2.3.2 Pitch Reset, Gradient Change

When analyzing pitch tracks (and the initial and final intonation at extraction points), it can be observed that speakers tend to jump to higher fundamental frequency values at the start of a new sentence. This phenomenon, which is called *pitch reset* (pr), is even more distinct when a paragraph (or in broadcast issues, a new topic) arises. It may also go along with a *gradient change* (gc) (section 2.3.1, figure 2.4).
These features (specified in [Hz]) can be directly gathered out of the final and initial intonation at detected extraction points by means of subtraction.

[4] The region for trend line analysis can be adjusted by tuning the parameter *Settings.range*.

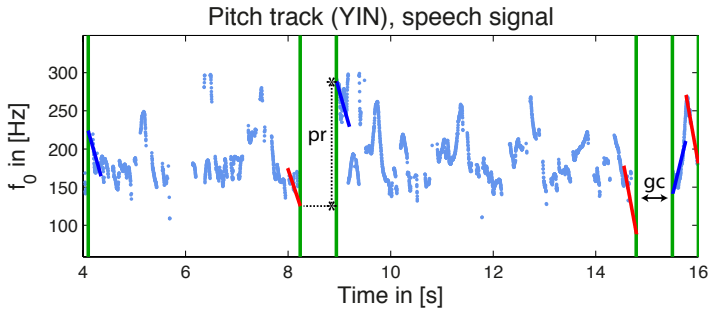

Figure 2.4: Pitch reset (pr) and gradient change (gc) at potential sentence boundaries;
light blue: pitch track, blue: initial intonation, red: final intonation, green: potential sentence boundaries

2.3.3 Downdrift

A general decline of pitch within a sentence is also called *downdrift* (dd). An explanation for this pitch trend can be found in lacking air in the lungs towards the end of a phrase, but also speaker intention could be seen as a possible reason (figure 2.5).
Within the pitch track, this feature can be again analyzed by using linear approximation between potential sentence boundaries and is quoted in [Hz/s]. Usually, smaller sentence-like phrases go along with steeper decays.

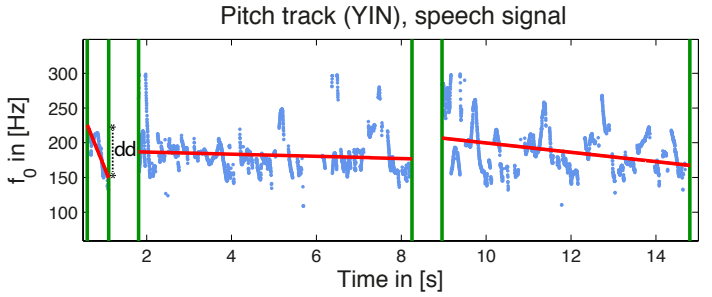

Figure 2.5: Downdrift (dd) within sentences with different durations;
light blue: pitch track, red: downdrift, green: potential sentence boundaries

To get a comparable quantity, this feature is normalized for decision tree processing:

$$Downdrift_{d,norm} = \frac{|Downdrift_d|}{\frac{1}{D}\sum_{d=1}^{D} Downdrift_d} \tag{2.2}$$

for all D (potential) sentences.

2.4 Feature Summary

In total, we end up with *three* energy-based and *five* pitch-related features which are summarized in table 2.1[5]. The behavior (and the importance in the decision process) of some features can be altered via tunable parameters. Section 4 explains the combination of these features and parameters in a decision tree.

Table 2.1: Summary of features and tunable parameters

	Feature	**Tunable parameter**
energy-based	Pause length	*MinPause*
	Sentence length	*MinSentenceLength*
	Energy decay	*MinDecay*
pitch-related	Initial intonation	-
	Final intonation	-
	Pitch reset	*MaxAlt*
	Gradient change	-
	Downdrift	*MaxAlt*

[5] All features are stored in a MATLAB structure array called *Features*.

3 The Algorithm

This section describes the elementary stages of the implemented algorithm (figure 3.1). First of all, the input signal $s[n]$ is split into Hanning-windowed half-overlapping frames with a length of N samples (see footnote 2, page 3).
For the subsequent VAD algorithm (described in detail in section 3.1), it is necessary to calculate the frame-based energy track, which is then used as input stream. The output of the VAD identifies voiced segments within the input signal excerpt $s'(n)$. It is not just consulted to determine the extraction points for further calculations of prosodic features by means of pause-length analysis, but also introduces first sentence boundaries in case of suspiciously long pauses[6]. Additionally, the already mentioned energy-based features (*PauseLength, SentenceLength, EnergyDecay*, section 2.2) can be calculated. Another advantage is the reduction of complexity, as the fundamental frequency estimator (section 3.2) merely deals with 'active', i.e. voiced frames.
The resulting pitch track comprises the origin for all pitch-related prosodic feature computations (*InitialIntonation, FinalIntonation, PitchReset, GradientChange, Downdrift*, section 2.3). As we are dealing with a f0-estimation, we use a slightly different constraint for the frame size: a signal's fundamental frequency can be reliably obtained when the analyzed excerpt contains at least three periods of the signal; thus the minimal frame length is determined by the lowest frequency to be tracked by the YIN-algorithm (section 3.2.1) with unchanged hop size between the frames.
All the obtained features are then combined in a decision tree (section 4) and processed to determine the correct sentence boundaries.

3.1 Voice Activity Detection

Voice activity detection (VAD) can be realized in several forms ranging from energy-based approaches [MK02] to algorithms in the spectral [PSJ$^+$02] and cepstral domain [HM93] just to name a few.
In this project thesis, an energy-based VAD approach with an adaptive threshold based on energy dynamics is used to classify speech into 'active' (voiced) and 'inactive' (unvoiced) segments of a continuous speech stream. Subsequently, the short-time VAD is smoothed and the potential extraction points (section 2.1) or sentence boundaries are determined. Benefits of the algorithm are sufficient functionality and low computational costs. Other main advantages lie in the fact that the algorithm can be implemented easily and is not topic or language dependent, but rather based on signal properties [PR09]. Robustness on a high level can be achieved by tuning few parameters.

[6] i.e. pauses whose length exceed the user parameter *MinPause*.

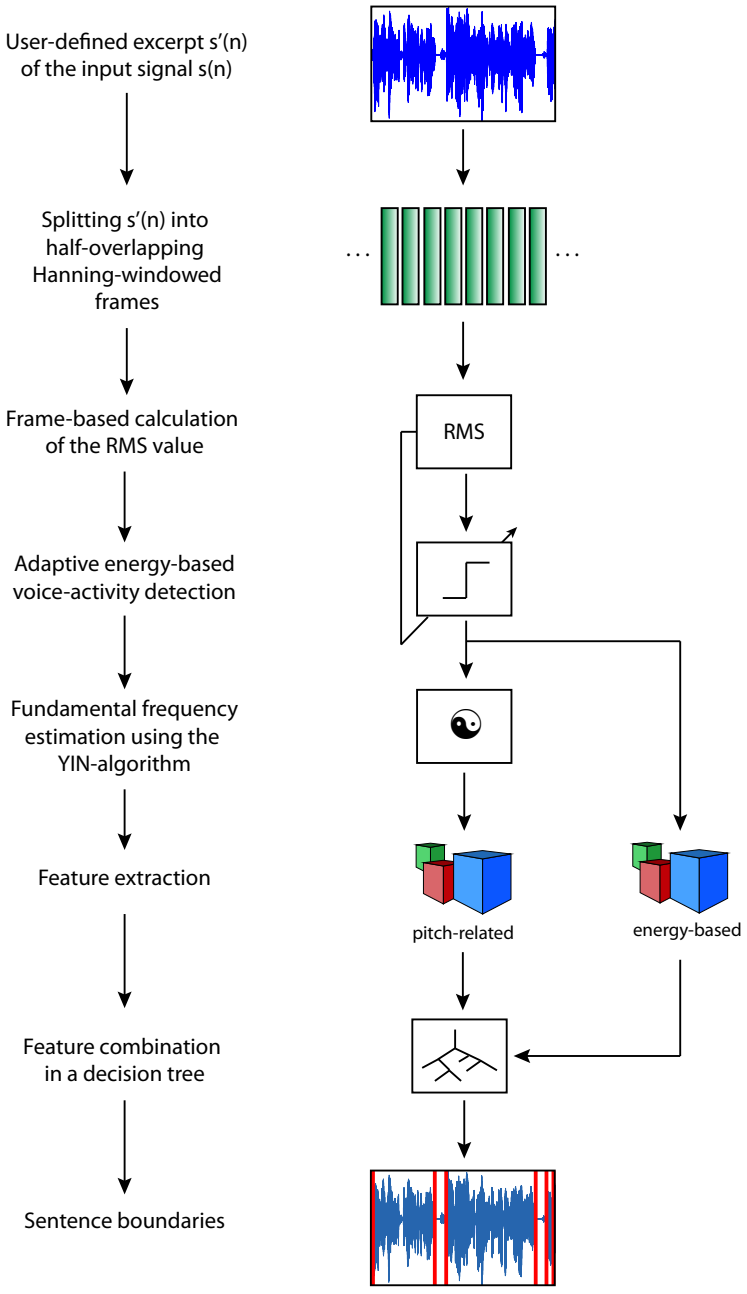

Figure 3.1: Functional blocks of the implemented algorithm

The VAD algorithm consists of two main stages: In the first stage, a frame-based short-time VAD is applied. In the second stage (post-processing), we use long-time frames by smoothing the initial detection output.

3.1.1 Basic Voice Activity Detection

To go into detail, the input signal firstly is split into frames of length 20 ms which are then Hanning-windowed and arranged half-overlapping[7]. The RMS-values can be easily calculated by using the relationship

$$E(i) = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} s[i \cdot M + n]^2} \tag{3.1}$$

where $E(i)$ denotes the root mean square value of the i-th frame, N is the frame size and M the hop size between concatenated frames (figure 2.1, page 4, upper plot).
From the value of this parameter, we can set a threshold to analyze a signal with respect to voice activity. In general, short-term speech activity is defined as

$$\mathit{VAD}(i) = \begin{cases} 1, & \text{if } E(i) \geq E_{thr}(i) \\ 0, & \text{if } E(i) < E_{thr}(i). \end{cases} \tag{3.2}$$

Since a fixed threshold mostly leads to unsatisfying results with too many short-term errors in such VAD definitions, an adaptive one based on updated RMS dynamics is proposed, i.e.

$$E_{thr}(i) = E_{min}(i) + \frac{p}{100} \cdot \underbrace{(E_{max}(i) - E_{min}(i))}_{\text{energy dynamics}}, \tag{3.3}$$

with p comprising the dynamic amount which should be added to the adaptively set minimum RMS power. The values of all parameters used in the VAD algorithm are listed in table 3.1.
Both $E_{max}(i)$ and $E_{min}(i)$, respectively, are updated based on exponential averaging:

$$E_{max}(i) = \begin{cases} q_{max1}E_{max}(i-1) + (1-q_{max1})E(i) \\ \qquad\qquad \text{if } E(i) \geq E_{max}(i-1), \\ \\ q_{max2}E_{max}(i-1) + (1-q_{max2})E(i) \\ \qquad\qquad \text{if } E(i) < E_{max}(i-1), \end{cases} \tag{3.4}$$

[7] A hop size of 10 ms ensures the tracking of the shortest phonemes (plosives) [Kuw96].

$$E_{min}(i) = \begin{cases} q_{min1}E_{min}(i-1) + (1-q_{min1})E(i) \\ \qquad\qquad \text{if } E(i) \leq E_{min}(i-1), \\ \\ q_{min2}E_{min}(i-1) + (1-q_{min2})E(i) \\ \qquad\qquad \text{if } E(i) > E_{min}(i-1). \end{cases} \tag{3.5}$$

The constants q_{max1}, q_{max2}, q_{min1}, q_{min2} in (3.4) and (3.5) are responsible for the update speed within the adaptive process (used values are listed in table 3.1). Whereas there should be a fast adaption on new situations, the "forgetting"-process must be slower (q_{max1} and q_{min1} are commonly chosen smaller than their counterparts). The energy minimum E_{min} has to be updated particular slowly (see figure 3.2) [PR09].

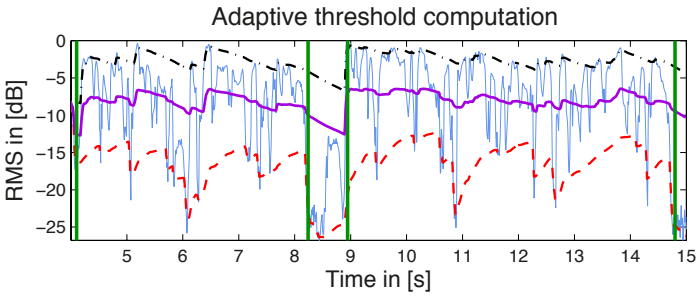

Figure 3.2: Adaptive threshold computation, $p = 25$;
light blue: intensity progression, black: E_{max},
red: E_{min}, magenta: E_{thr}, green: potential sentence boundaries

To overcome the problem that arises within speech pauses when energy dynamics get very small, a parameter E_{dmin} which requests minimum dynamics is introduced, so that noise (or breaths before a new sentence) is not classified as speech, meaning that

$$VAD(i) = 0, \quad \text{if } E_{max}(i) - E_{min}(i) < E_{dmin}. \tag{3.6}$$

3.1.2 Post-processing Stage

A typical problem of energy-based VAD algorithms is the over-alertness to very short speech parts or other short-time impulses within the signal. As we are interested in sentences with a certain utterance length (which is definitely longer than the frame size), these short-time errors can be effectively reduced by buffering the detected active frames into long-time frames. The buffer length T is chosen to be 0.1 s – a long-time buffer thus comprises $K = T/M$ frames. If the energy inside a buffer-frame exceeds a defined threshold value thr_{buff} (table 3.1), the

whole buffer will be defined as 'active'. To express these constraints for the m-th buffer, we can write

$$\mathit{buff}_m(j) = \mathit{VAD}(m \cdot K + j) \quad \text{for} \;\; j = 0, 1 \ldots K-1, \tag{3.7}$$

$$\mathit{VADbuff}_j = \begin{cases} 1, & \text{if} \;\; \dfrac{1}{K} \displaystyle\sum_{j=0}^{K-1} \mathit{buff}_m(j) \geq \mathit{thr}_{buff} \\ \\ 0, & \text{if} \;\; \dfrac{1}{K} \displaystyle\sum_{j=0}^{K-1} \mathit{buff}_m(j) < \mathit{thr}_{buff}. \end{cases} \tag{3.8}$$

With the results of this analysis process (figure 3.3), we can use the long-time buffer to set extraction points as candidates for sentence boundaries.

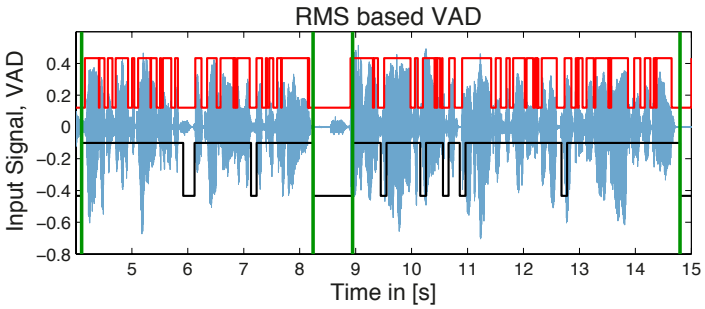

Figure 3.3: Detection process with post-processing steps;
light blue: input signal excerpt, red: short-time VAD,
black: long-time VAD, green: start-/end-point detection

It is reasonable to establish a minimum utterance length[8] (which could be just one word) and extract the beginning of it with the help of the following definition:

$$\mathit{VAD}_j = \mathit{VADbuff}_j \quad \text{if} \;\; \mathit{VADbuff}_j = 1, \tag{3.9}$$

hence the start point is defined by the start of the respective buffer frame.

The end-point detection goes along with a minimum pause length[9] whose preset value can also be altered by the user (table 3.1). Once an "active" frame has been detected, subsequent long-time buffers are set 'active' until the minimum of P buffers, defined by the minimum pause length (typically between 0.4 - 1.5 s), is reached, i.e.

[8] The user is able to tune the parameter *MinSentenceLength* [s] to ignore too short utterances.
[9] By tuning the parameter *MinPause* [s], inter-word pauses are disregarded as they are much shorter than pauses that separate two utterances.

$$\begin{aligned} VAD_j = 1, \quad & \text{if } VAD_{j-1} = 1, \\ VAD_{j-k} = 0, \quad & \text{for } k = 0, 1 \dots P-1 \\ & \text{if } VADbuff_{j-k} = 0. \end{aligned} \tag{3.10}$$

The detected boundaries are stored in a matrix with two columns containing the start- and end-points respectively with a discretization interval defined by the buffer length T. Start-points are dedicated to the buffer start, while end-points coincide with the end of a buffer (figure 3.3).

It can be observed that the VAD algorithm is prone to detect the boundaries with a certain delay. As a countermeasure, a "safe band" is introduced to shift the utterance enclosures a bit[10].

3.1.3 Chosen Settings for the VAD algorithm

Table 3.1 displays the range of all settings[11] used for the VAD algorithm which were determined experimentally. At least, the parameters *MinSentenceLength* and *MinPause* have to be tuned speaker-related. A full listing of all values can be found in appendix C.

Table 3.1: Chosen settings for the VAD algorithm

Type	Parameter	Range / Value
Settings	*SafePre*	0 – 0.1 s
	SafePost	0.05 – 0.1 s
	MinSentenceLength	0.1
	MinPause	0.48 – 0.7
Advanced	q_{min1}	0.5
	q_{min2}	0.9989
	q_{max1}	0.7
	q_{max2}	0.99
	p	25
	E_{dmin}	6 dB
	t_{anal}	0.1 s

[10] For this purpose, two parameters, *SafePre* and *SafePost*, are introduced and allow an independent boundary disclosure of start- and end-points (table 3.1).

[11] All settings are stored in two MATLAB structure arrays: *Settings* for basic settings and *Advanced* for advanced settings, each of them being predefined by default values. The *VADprosody.m* help file can be consulted for further information.

3.2 Fundamental Frequency Tracking

Pitch is defined as the perceived fundamental frequency of a periodic signal which can be obtained by taking the inverse of its period ($f_0 = 1/T$). This is true as long as a sound is perfectly periodic, but the results of pitch detection algorithms (PDA) vary significantly when applied to real-life signals (speech or music) with time-varying and non-stationary characteristics [BSH07, vdKZ10]. In speech signals, the fundamental frequency is produced in the glottis and can be equated with the rate of vocal fold vibration (which is not perfectly periodic because of movements of the vocal tract and certain superimposed aperiodicities [dCK02]).
Since a robust fundamental frequency (f_0) estimation algorithm is essential for a reliable derivation of pitch-related prosodic features, several pitch detection approaches have been studied in-depth. A realization is possible in time- or lag-domain (using auto-correlation functions [Boe93]) as well as in the spectral [ZH08] or cepstral domain [AS99] not to forget approaches based on auditory models [CPZ98]. Comparing the results in [vdKZ10], the YIN-algorithm (based on a modified autocorrelation), developed by de Cheveigné and Kawahara [dCK02] seemed to be the most promising.

3.2.1 The YIN-Algorithm

The YIN-algorithm (deduced from yin and yang to symbolize interaction between autocorrelation and its triggered partial cancellation) used in this project thesis is based on an autocorrelation function but improved by several modifications [dCK02]. Just a few parameters have to be set by the user, what makes this algorithm very robust against different speakers and speaking styles.
In this section, the implemented algorithm with all its modifications and enhancements is described step by step. The notation used by the authors will be maintained.

Step 1: Difference Function The definition for a slightly modified autocorrelation function (ACF) $r_t(\tau)$ at time index t with window size W

$$r_t(\tau) = \sum_{j=t+1}^{t+W} x_j x_{j+\tau} \tag{3.11}$$

has the advantage not to decrease in comparison to the standard notation of the ACF, where the window W gets smaller when increasing the lag τ. If we take a periodic input signal and insert it into equation (3.11), we can choose the first non-zero-lag peak to compute the fundamental frequency by taking its inverse. Herein lies the fundamental problem of autocorrelation-based approaches, because the algorithms are prone to select either the zero-lag or too high lags in case of wrong adjustment of range parameters, which inevitably leads to so-called *octave errors*.

Therefore, the authors of [dCK02] suggest a different approach to obtain the period T within a periodic signal, the already mentioned *difference function*

$$d_t(\tau) = \sum_{j=1}^{W} (x_j - x_{j+\tau})^2. \tag{3.12}$$

The roots of this function determine the signal period τ and its integer multiples, including the zero-lag (figure 3.4, upper plot).

Step 2: Cumulative Mean Normalized Difference Function Unless there are no range restrictions for the minimum retrieval, the algorithm will always choose the undesired zero-lag. To overcome this drawback, a slightly modified difference function is introduced, the *cumulative mean normalized difference function* (CMNDF)

$$d'_t(\tau) = \begin{cases} 1, & \text{if } \tau = 0, \\ d_t(\tau) \Big/ \left[\frac{1}{\tau} \sum_{j=1}^{\tau} d_t(j) \right], & \text{otherwise.} \end{cases} \tag{3.13}$$

The values of the difference function $d_t(\tau)$ (equation 3.12) are divided by some sort of average value that is determined by the lag τ. This approach has some beneficial effects: $d'_t(\tau)$ starts at the value "1" and remains high-valued in lower lag regions, until it drops below the average and defines the period by the lowest dip in the CMNDF (figure 3.4, lower plot). As a consequence, the need for a restriction of the upper frequency search range to avoid the zero-lag dip can be disregarded.

Step 3: Absolute Threshold In addition to the normalization of the CMNDF in step 2, there is another improvement to refine the algorithm's error rate. The CMNDF may show a global minimum at higher-order dips within the search range. Octave errors due to this phenomenon can be reduced by the introduction of a small threshold value and to take the first lag τ that falls below it. The authors suggest a threshold value of 0.1 (figure 3.4, green dashed line in the lower plot).

Step 4: Parabolic Interpolation So far, it is possible to detect a signal period that is a multiple of the sampling period. This means that the dip may be displaced up to half the sampling period, which can lead to unacceptable pitch errors (*gross errors*). To achieve sub-sample accuracy, it is thus necessary to apply a parabolic estimation (appendix B) with subsequent selection of the accordant minimum (value of the abscissa).

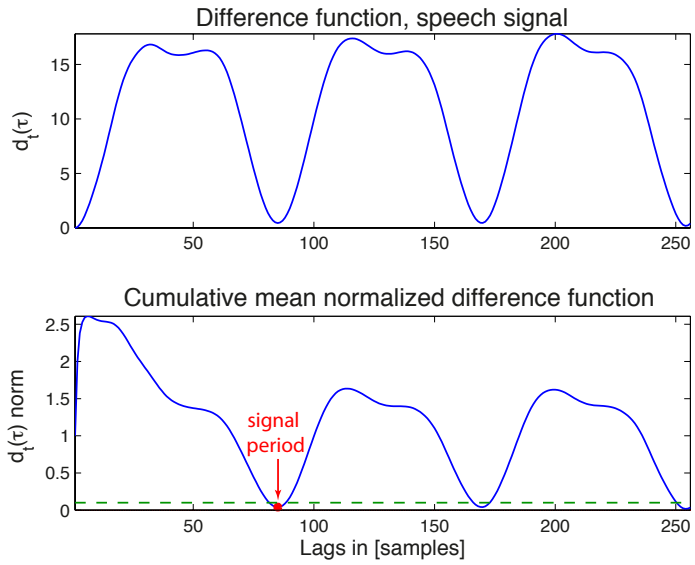

Figure 3.4: Above: difference function,
Below: cumulative mean normalized difference function;
green: absolute threshold, red dot: detected signal period

Step 5: Best Local Estimate Up to this point, the YIN-algorithm sometimes delivers strongly fluctuating pitch values (i.e. mostly too-high values). To obtain a stable pitch track, a largest expected time lag T_{max} which defines a refined search interval is introduced as a countermeasure. More precisely, the algorithm searches a minimum of $d'_\theta(T_\theta)$ for θ at time t within the interval $[t - T_{max}/2, t + T_{max}/2]$. The authors use a value of $T_{max} = 25$ ms for this 'dynamic programming' approach and affirm an additional error rate reduction.

Applying Step 1-5 leads to satisfying results when comparing the YIN-algorithm to other fundamental frequency estimators [dCK02, vdKZ10]. In addition to, the results can be achieved fast by tuning just a few parameters and the occurrence of octave errors is quite rare.

3.2.2 Chosen Parameters for the YIN Algorithm

Table 3.2 displays the most important parameters[12] used for the YIN algorithm which were determined experimentally by selecting sample files from the "Aix-MARSEC" library.

Table 3.2: Chosen parameters for the YIN algorithm

Gender	Parameter	Value
female	*minf0*	100 Hz
	maxf0	290 Hz
	thresh	0.1
male	*minf0*	80 Hz
	maxf0	250 Hz
	thresh	0.1

[12] All YIN parameters are stored in a MATLAB structure array P. It is possible to tune additional parameters. The *yin.m* help file (in the YIN directory) can be consulted for further informations.

4 Decision Tree Processing

4.1 Structure of the tree

All found prosodic features are combined in an experimentally determined *binary decision tree* which consists of *three stages*. Tests with independent feature usage resulted in a feature hierarchy that is represented by the structure of the tree. The individual stages comprise the following type of features:

- **STAGE 1: *basic features*:** *MinSentenceLength*, *MinPause*
- **STAGE 2: features related to *interior structure*:** *EnergyDecay*, *Downdrift*
- **STAGE 3: features related to *boundary intonation*:** *FinalLowering*, *PitchReset*, *GradientChange*.

As top features, *MinSentenceLength* and *MinPause* became tangible and thus make up STAGE 1 within the tree. In this first step, not only too short utterances but also negligibly short pauses (inter-word pauses) are ignored in the input signal excerpt. Additionally, an *exit criterium* is introduced comprising the occurrence of conspicuously long pauses which exceed a length of 0.7 s (the value has been determined experimentally). All involved boundaries are *locked* and no longer can be erased in the consequent stages of the tree (figure 4.1 with details for STAGE 1).

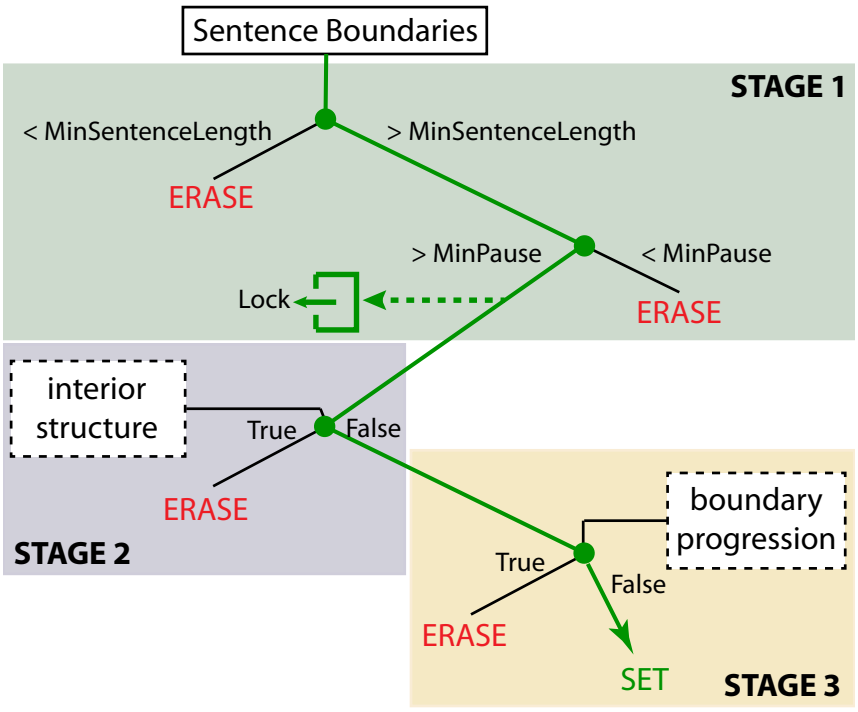

Figure 4.1: Structure of the used decision tree: details for STAGE 1

After this basic processing, the remaining sentence boundaries are passed on to STAGE 2 (figure 4.2, light-lavender shaded area). In this part of the tree, features that are related to the interior sentence structure are evaluated. The combination is realized through logical connectives (conjunction and disjunction).

To go into detail, the normalized energy decay (*EnergyDecay*$_{d,norm}$, equation 2.1, page 4) and the normalized downdrift (*Downdrift*$_{d,norm}$, equation 2.2, page 7) of the d-th sentence, respectively, is compared to the tunable parameter *MaxAlt*[13]. If one of these features exceeds *MaxAlt* the OR gate will return a TRUE. This condition should include the observed phenomenon of excessive pitch and intensity variation within (too) short utterances.

It has become manifest through experiments that the combination of rather less reliable features (features of STAGE 2 and 3) with highly trustworthy ones (features of STAGE 1) leads to better results. So the output of the OR gate is linked to an additional AND gate with the feature *SentenceLength* being compared to a parameter *thr1*[14] as second input.

As a result, only the sentence boundaries (not being locked in STAGE 1) that fulfill one of the two conditions are erased within this stage, the others are handed on to STAGE 3.

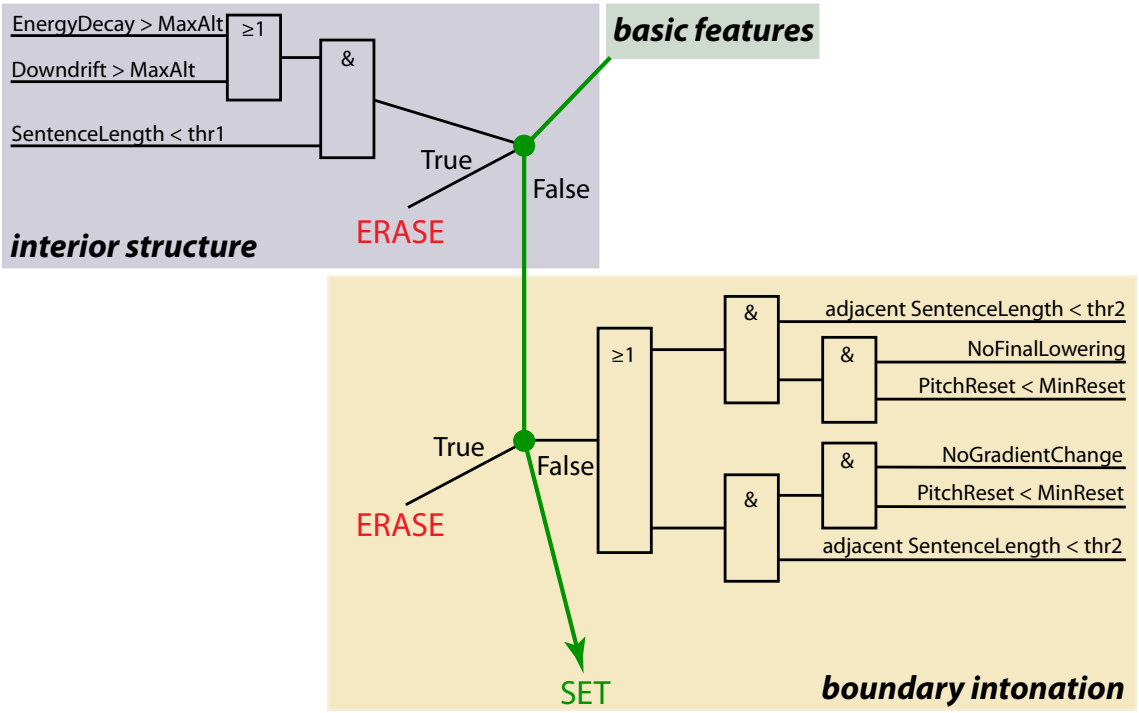

Figure 4.2: Structure of the used decision tree: details for STAGE 2 and 3

[13] All decision tree-related parameters are stored in the MATLAB structure array *Advanced*.
[14] See footnote 13.

The final stage of the decision tree (figure 4.2, light-amber shaded area) primarily deals with boundary-related pitch features. To achieve a higher reliability of the found features, the combination is executed in a more stringent way – features are almost exclusively combined by means of AND gates.
In case of a detected true sentence boundary, a final lowering of pitch values and a pitch reset that exceeds the tunable parameter *MinReset*[15] is expected. This is implemented by conjuncting the features *FinalIntonation* and *PitchReset*.
A similar approach is chosen for the second branch within this stage: for a true sentence boundary, there is a demand for a gradient change and a pitch reset exceeding *MinReset* (the accordante features *GradientChange* and *PitchReset* are again linked via an AND gate).
To raise reliability, it is once more reasonable to connect these two conditions with a basic feature – the *SentenceLength* of adjacent utterances (exceeding the second threshold value *thr2*[16]). If one of the two described branches turn out to be TRUE (disjunction) the involved sentence boundaries will be erased.

Finally, we obtain assessed sentence boundaries which are much more trustworthy than those that were detected during the analysis process before the decision tree.

4.2 Chosen Parameters for the Decision Tree

The range of all experimentally found tunable parameters can be found in table 4.1.

Table 4.1: Chosen advanced parameters for the decision tree

Type	Parameter	Range
Advanced	*MaxAlt*	8 – 10 dB
	MinReset	70 – 100 Hz
	thr1	0.5 – 3 s
	thr2	0.7 – 2.5 s

[15] See footnote 13.
[16] See footnote 13.

5 Evaluation

To evaluate the algorithm under various conditions (gender, speaking style...), we choose 6 different speakers of the Aix-MARSEC database (four male and two female speakers, respectively) who should be representative for the whole database.
The algorithm is tuned by setting the most important speaker-related parameters (VAD settings: tables 3.1, section C.1, YIN settings: table 3.2 and decision tree settings in table 4.1). The algorithm is optimized in a way that rather too many boundaries than too less are detected – obviously leading to some additional *false positives*.
The corresponding label files (which contain hand-labelled sentence boundaries) of the "Aix-MARSEC" database are loaded and compared with found sentence boundaries. The label files also determine the beginning and the end of the signal excerpts by searching the first and the last sentence boundary sign (||) and applying the found times to *Settings.ExcerptStart* and *Settings.ExcerptEnd*, respectively.

5.1 Definitions for the Evaluation

To facilitate the notation for the system performance analysis, it is necessary to introduce some symbols (similar to [MKSW99]), i.e.

$TD =$ **totally detected**
total number of hand-labelled sentence boundaries

$TP =$ **true positives**
number of correctly detected sentence boundaries that lie
in a pre-defined area around the manually labelled boundaries ($\pm$ 100 ms)

$FP =$ **false positives**
number of incorrectly detected sentence boundaries that lie
outside all pre-defined areas around the manually labelled boundaries

$FN =$ **false negatives**
number of hand-labelled sentence boundaries not found by the algorithm.

Using these symbols, we can derive some significant evaluation measurements:

$$P = \frac{TP}{TP + FP} \cdot 100 \ \ [\%] \ldots \textbf{precision}, \tag{5.1}$$

$$R = \frac{TP}{TP + FN} \cdot 100 \ \ [\%] \ldots \textbf{recall}. \tag{5.2}$$

While *precision* gives information about how many of the detected boundaries are true, the measurement *recall* tells us how many true boundaries have been found.

These values can be combined to obtain the algorithm's efficiency with reference to its hit rate. The so-called *F-measure*, a metric to rate the system performance by a single value, is defined as the weighted harmonic mean of P and R [MKSW99]:

$$F = \frac{2 \cdot P \cdot R}{P + R} \cdot 100 \ \ [\%] \ldots \textbf{F-measure}. \tag{5.3}$$

For the sake of completeness, there is another popular measurement to assess the algorithm's performance:

$$S = \frac{TP}{TP + FN + FP} \cdot 100 \ \ [\%] \ldots \textbf{score}. \tag{5.4}$$

5.1.1 Evaluation Note

To put the evaluation into perspective, it is fair to note that not all *false positives* are based on imprecisions of the algorithm. In fact, some sentence boundaries were labelled quite inexactly. To undermine this drawback, figure 5.1 should serve as an example.

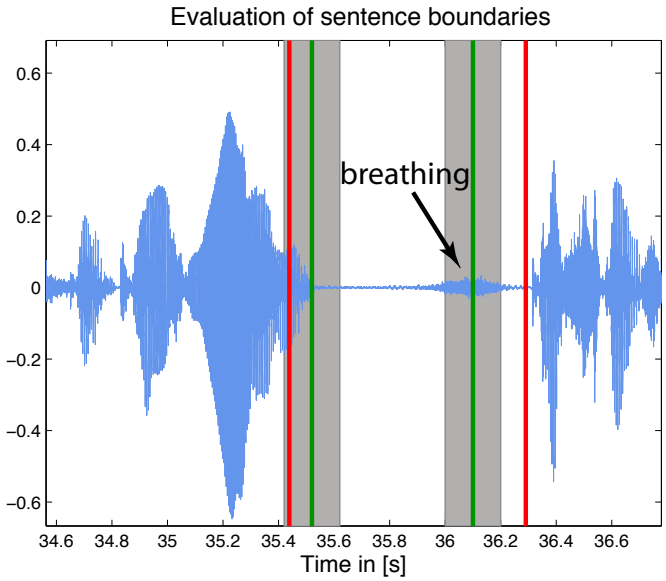

Figure 5.1: Deflection of hand-labelled and auto-detected sentence boundaries

green: hand-labelled sentence boundaries, red: auto-detected sentence boundaries, light blue: input signal excerpt

5.2 Speaker-related Results for the Aix-MARSEC Database

In total, 18min35s of speech containing 407 sentences were analyzed and evaluated. Table 5.1 lists the values of all involved symbols and evaluation quantities.

Table 5.1: Speaker-related evaluation results, $\Delta = \pm 100$ ms

Speaker (Gender)	TD	TP	FP	FN	P [%]	R [%]	F [%]	S [%]
A01 (f)	88	70	18	2	79.8	97.6	87.2	77.7
A03 (m)	76	52	20	4	75.7	93.8	83.3	72.6
A05 (m)	108	88	24	6	76.5	95	84.1	73.0
A11 (m)	62	45	31	0	58.9	100	73.4	58.9
C01 (m)	408	308	128	12	71.1	96.2	80.3	69.1
H04 (f)	72	54	28	6	68.8	91.7	76.3	61.9
Sum / Mean	**814**	**617**	**249**	**30**	**71.8**	**95.7**	**80.8**	**68.9**

The last row contains the sum and the mean of all involved quantities and gives information about the overall performance of the algorithm. It becomes evident that the algorithm has some drawbacks with respect to *Precision* and *Score*, respectively. This fact is certainly based on three main reasons:

1. **Tuning aspect:** the algorithm is tuned to detect rather too many existing boundaries which leads to a higher quota of *false positives*.
2. **Inexact hand-labelling:** cf. section 5.1.1 and figure 5.1.
3. **Discretization of sentence boundaries:** The VAD (section 3.1, page 9) only delivers sentence boundaries discretely arranged on a time grid with 100 ms steps which sometimes are outside the tolerance area ($\Delta = \pm 100$ ms).

Nevertheless, the evaluation is satisfying when looking at the good *F-measure* (> 80 %) results and very good results concerning *Recall* (> 95 %), which means that the better part of all existing sentence boundaries have been detected.

6 Summary and Conclusion

This project thesis suggests a simple but robust algorithm for automatic sentence segmentation of speech relying upon prosodic features. The two main functional blocks during the analysis process – the adaptive, energy-based voice activity detection and the fundamental frequency estimator (YIN) – both can be implemented with low computational costs and are optimized with the help of just a few parameters. Noticeable error rate reduction is achieved by a combination of the found prosodic features in a decision tree which leads to assessed and much more reasonable sentence boundaries.

Concerning functionality, the algorithm in the current state is only applicable for "well conditioned" speech (recited texts) but cannot analyze spontaneous spoken word because of manifold reasons (filled pauses, fillers, false starts, repetitions, hesitations...) and is not adapted for speaker turns (dialogs, interviews...) or background noise. Languages similar to British English (which was used for test and evaluation) may be compatible as long as the discussed prosodic features may be utilized analogously.

The algorithm can be used for rough pre-labelling of long recordings or speech databases. For refinement, it is recommended to examine the found sentence boundaries, erase erroneously chosen ones and render the correct boundaries more precisely (application dependent).

Improvements of the algorithm's overall performance can be achieved by using a more precise voice activity detection that delivers potential sentence boundaries on a more acute time grid. Furthermore, the embracement of duration features (syllable-based features, pre-boundary lengthening...) that affect the speaking rate as well as other features like speaker turns or speaker overlaps would augment the application area of the algorithm. Another challenge lies in the implementation of a pre-processing noise estimation and reduction stage to overcome the problem of time-varying background noise.

Future works may also concern the real-time analysis of prosodic features within sentence boundaries to analyze emotion, stress or workload of a speaker in critical working environments with a high level of responsibility.

Appendix

A Linear Approximation

In linear approximation, a set of given data points y_i is estimated by a straight line that optimally represents a tendency. This approach finds application in several different scientific fields like statistics, finance, economics or epidemology and usually is used as a meaningful tool for trend line computation.
With a given data set $\{x_i, y_i\}$ ($i = 1, 2, 3 \dots R$), the goal is to find a simple first-order polynom defined by the equation

$$\psi = kx_i + d \tag{A.1}$$

and to estimate the parameters k and d, respectively.

To minimize the sum of squared residuals (equation A.2), we use a quadratic approach; i.e. the error is optimized in the *least-square* (LS) *sense*. The error samples of the regression model

$$e_i = \sum_{i=1}^{R} (y_i - \hat{y}_i)^2 \tag{A.2}$$

are defined by the difference between the data points y_i and the estimated data points $\hat{y}_i$. In vector / matrix notation we can write

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_R \end{pmatrix}_{[Rx1]} , \quad \hat{\boldsymbol{y}} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_R \end{pmatrix}_{[Rx1]} , \quad \boldsymbol{X} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_R & 1 \end{pmatrix}_{[Rx2]} , \quad \boldsymbol{\alpha} = \begin{pmatrix} k \\ d \end{pmatrix}_{[2x1]} \tag{A.3}$$

with x_i marking the range of the straight line and $\boldsymbol{\alpha}$ containing the optimal coefficients (slope and offset) to be calculated.

The error function now takes the form $\|\mathbf{e}\| = \|(\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})\|$, with $\mathbf{y}$ comprising the values that should be estimated. Consequently, a cost function is derived for further computations, i.e.

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \mathbf{e}^T\mathbf{e} = (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}) = \\ &= \mathbf{y}^T\mathbf{y} - (\mathbf{X}\boldsymbol{\alpha})^T\mathbf{y} - (\mathbf{X}\boldsymbol{\alpha})^T\mathbf{y} + \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha}. \end{aligned} \tag{A.4}$$

The next step to receive the optimal coefficients $\boldsymbol{\alpha}_{LS}$ is to calculate the gradient of the cost function, which is set to zero afterwards:

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}}\{J(\boldsymbol{\alpha})\} &= \frac{\partial}{\partial \boldsymbol{\alpha}}\{J(\boldsymbol{\alpha})\} = \\ &= -2\mathbf{y}^T\mathbf{X} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \mathbf{0} \\ \mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} &= \mathbf{y}^T\mathbf{X} \\ \boldsymbol{\alpha}_{LS} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \end{aligned} \tag{A.5}$$

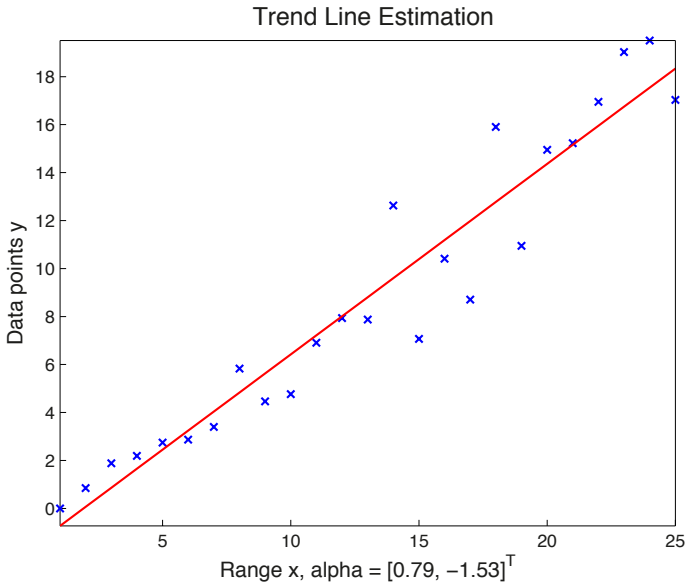

Figure A.1: Illustrative example for an optimal trend line estimation using least-squares

B Parabolic Interpolation

With three given abscissae $\{x_1, x_2, x_3\}$ and their corresponding merits $\{f(x_1), f(x_2), f(x_3)\}$ we can apply a parabola to find a minimum x_{min} that lies in the vicinity of x_2. It can be calculated by using the relationship [Del99]

$$x_{min} = x_2 - \frac{1}{2} \cdot \frac{(x_2 - x_1)^2[f(x_2) - f(x_3)] - (x_2 - x_3)^2[f(x_2) - f(x_1)]}{(x_2 - x_1)[f(x_2) - f(x_3)] - (x_2 - x_3)[f(x_2) - f(x_1)]}. \tag{B.1}$$

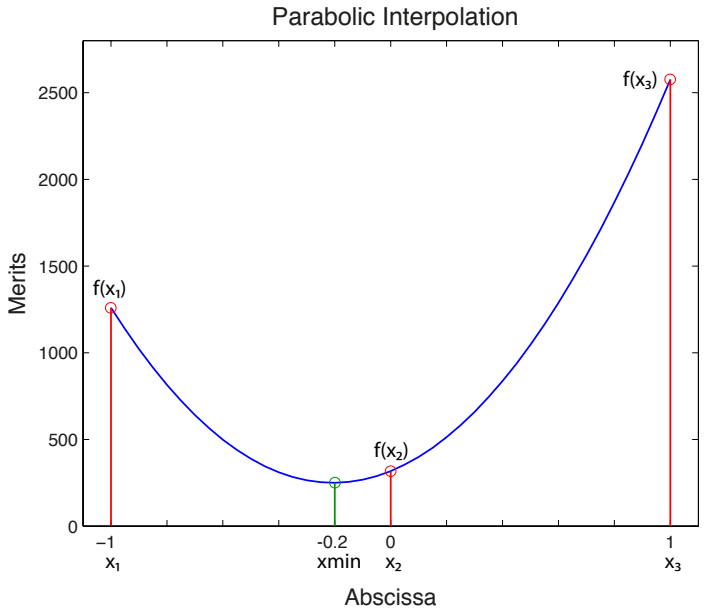

Figure B.1: Finding an exact minimum by parabolic interpolation

C Entire List of VAD-related Parameters

Table C.1: Speaker-related VAD settings

Speaker (sex)	Settings	Value
A01 (f)	*MinSentenceLength [s]*	0.1
	MinPause [s]	0.5
	SafePre [s]	0.1
	SafePost [s]	0.1
A03 (m)	*MinSentenceLength*	0.1
	MinPause [s]	0.6
	SafePre [s]	0.1
	SafePost [s]	0.1
A05 (m)	*MinSentenceLength*	0.1
	MinPause [s]	0.55
	SafePre [s]	0
	SafePost [s]	0.05
A11 (m)	*MinSentenceLength*	0.1
	MinPause [s]	0.48
	SafePre [s]	0
	SafePost [s]	0
C01 (f)	*MinSentenceLength*	0.1
	MinPause [s]	0.7
	SafePre [s]	0
	SafePost [s]	0.05
H04 (f)	*MinSentenceLength*	0.1
	MinPause [s]	0.48
	SafePre [s]	0
	SafePost [s]	0.06

References

[ABH04] C. Auran, C. Bouzon, and D. Hirst, "The Aix-MARSEC project: an evolutive database of spoken British English," CNRS UMR 6057, Laboratoire Parole et Langage Université de Provence, Aix-en-Provence, France, Tech. Rep., 2002-2004.

[AS99] S. Ahmadi and A. S. Spanias, "Cepstrum-based pitch detection using a new statistical v/uv classification algorithm," *Ieee Transactions On Speech And Audio Processing*, vol. 7, no. 3, pp. 333–338, 1999.

[Boe93] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings 17 (1993), 97-110*. Institute of Phonetic Sciences, University of Amsterdam, 1993.

[BSH07] J. Benesty, M. M. Sondhi, and Y. A. Huang, *Springer Handbook of Speech Processing*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[CHJC+06] K. Chen, M. Hasegawa-Johnson, A. Cohen, S. Borys, S.-S. Kim, J. Cole, and J.-Y. Choi, "Prosody dependent speech recognition on radio news corpus of american english," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 232 – 245, Jan 2006.

[CPZ98] P. Cosi, S. Pasquin, and E. Zovato, "Auditory modeling techniques for robust pitch extraction and noise reduction," in *Proc. of ICSLP (in press*, 1998, pp. 105–3.

[DC11] G. Damnati and D. Charlet, "Robust speaker turn role labeling of tv broadcast news shows," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, may 2011, pp. 5684 –5687.

[dCK02] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[Del99] M. Delaurenti, "Design and optimization techniques of high–speed VLSI circuits," Ph.D. dissertation, Politecnico di Torino, Portineria 10129 Torino, Corso Duca degli Abruzzi, 24, Dec 1999.

[GFCP07] C. Gonzalez-Ferreras and V. Cardeoso-Payo, "A system for speech driven information retrieval," in *Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on*, dec. 2007, pp. 624 –628.

[HM93] J. Haigh and J. Mason, "Robust Voice Activity Detection Using Cepstral Features," in *TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference*, no. 0, Oct 1993, pp. 321 –324 vol.3.

[HPW04] K. Hacioglu, B. Pellom, and W. Ward, "Parsing speech into articulatory events," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, vol. 1, may 2004, pp. I – 925–8 vol.1.

[Jua10] C. Juan, "Research and implementation english morphological analysis and part-of-speech tagging," in *E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on*, vol. 2, april 2010, pp. 496 –499.

[KD06] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York: Springer, 2006.

[KFV+03] S. Kajarekar, L. Ferrer, A. Venkataraman, K. Sonmez, E. Shriberg, A. Stolcke, H. Bratt, and R. Gadde, "Speaker recognition using prosodic and lexical features," in *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, nov.-3 dec. 2003, pp. 19 – 24.

[Kuw96] H. Kuwabara, "Acoustic properties of phonemes in continuous speech for different speaking rate." in *ICSLP*. ISCA, 1996.

[LD98] V. Lecuit and D. Demolin, "The Relationship Between Intensity and Subglottal Pressure with Controlled Pitch," Laboratoire de Phonologie Expérimentale, Université Libre de Bruxelles CP 175, 50 av. F.-D. Roosevelt, 1050 Brussels, Belgium, Tech. Rep., 1998.

[Lli92] J. Llisterri, "Speaking styles in speech research," in *ELSNET/ESCA/SALT Workshop on Intergating Speech and Natural Language*, Dublin, Ireland, Jul. 1992, pp. 17–37. [Online]. Available: http://liceu.uab.es/~{}joaquim/publicacions/SpeakingStyles_92.pdf

[MK02] M. Marzinzik and B. Kollmeier, "Speech pause detection for noise spectrum estimation by tracking power envelope dynamics," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 109–118, 2002.

[MKSW99] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *In Proceedings of DARPA Broadcast News Workshop*, 1999, pp. 249–252.

[PL97] G. P.M. and Laan, "The contribution of intonation, segmental durations, and spectral features to the perception of a spontaneous and a read speaking style," *Speech Communication*, vol. 22, no. 1, pp. 43 – 65, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639397000125

[PR09] P. Pollák and R. Rajnoha, "Long Recording Segmenation Based on Simple Power Voice Activity Detection with Adaptive Threshold and Post-Processing," *SPECOM'2009, St. Petersburg*, 21-25 June 2009.

[PSJ+02] R. Prasad, A. Sangwan, H. S. Jamadagni, M. Chiranth, R. Sah, and V. Gaurav, "Comparison of voice activity detection algorithms for voip," in *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, ser. ISCC '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 530–.

[SSHTT00] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Commun.*, vol. 32, pp. 127–154, September 2000.

[SSRS01] R. Singh, M. Seltzer, B. Raj, and R. Stern, "Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, pp. 273–276, 2001.

[vdKZ10] A. von dem Knesebeck and U. Zölzer, "Comparison of pitch trackers for real-time guitar effects," in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, September 2010.

[WL01] G.-D. Wu and C.-T. Lin, "Noisy speech segmentation with multiband analysis and recurrent neural fuzzy network," in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 1, july 2001, pp. 540 –544 vol.1.

[WLL09] C.-H. Wu, C.-H. Lee, and C.-H. Liang, "Idiolect extraction and generation for personalized speaking style modeling," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 127 –137, jan. 2009.

[XYL+10] L. Xie, Y. Yang, Z.-Q. Liu, W. Feng, and Z. Liu, "Integrating acoustic and lexical features in topic segmentation of chinese broadcast news using maximum entropy approach," in *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, nov. 2010, pp. 407 –413.

[ZH08] S. Zahorian and H. Hu, "A spectral/temporal method for robust fundamental frequency tracking," *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4559–71, 2008.