SyncTS: Automatic synchronization of speech and text documents

David Damm[1], Harald Grohganz[1], Frank Kurth[2], Sebastian Ewert[1], and Michael Clausen[1]

[1]*University of Bonn, Department of Computer Science III, Römerstraße 164, 53117 Bonn, Germany*

[2]*Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), Neuenahrer Str. 20, 53343 Wachtberg, Germany*

Correspondence should be addressed to David Damm (`damm@cs.uni-bonn.de`)

ABSTRACT

In this paper, we present an automatic approach for aligning speech signals to corresponding text documents. For this sake, we propose to first use text-to-speech synthesis (TTS) to obtain a speech signal from the textual representation. Subsequently, both speech signals are transformed to sequences of audio features which are then time-aligned using a variant of greedy dynamic time-warping (DTW). The proposed approach is both efficient (with linear running time), computationally simple, and does not rely on a prior training phase as it is necessary when using HMM-based approaches. It benefits from the combination of a) a novel type of speech feature, being correlated to the phonetic progression of speech, b) a greedy left-to-right variant of DTW, and c) the TTS-based approach for creating a feature representation from the input text documents. The feasibility of the proposed method is demonstrated in several experiments.

1. INTRODUCTION

Textual information can be available in various forms and document types addressing different modalities such as plain text files (e.g., transcripts), scans of printed text (i.e., images), or spoken language (i.e., speech recordings). The merging and cross-form linkage (or alignment) of different documents embodying the same or at least—to some degree—similar content leads to new possibilities such as cross-modal search, merged presentation, and interlocked navigation. Therefore, the automated discovery of semantic cross-connections between different documents is an important and highly valuable task. In this paper, we deal with the interrelation of plain text files and speech recordings, each of which describe the same abstract information, but on a different modality. The aim is then to find a temporal alignment—i.e., a cross-modal linkage of semantically equivalent document parts—between the documents.

We consider three application scenarios benefiting from such a linkage. First, the cross-modal linkage is utilized to play back text and an audio recording of the text in a karaoke-like style, i.e., to time-synchronously display single words corresponding to the currently played part of the audio. Second, we use the text representation for navigation purposes, i.e., scrolling through the text while automatically changing the audio playback position by selecting a word in the text document. Third, in a query-retrieval setting, a text-based search not only provides the exact places of occurrence within a particular text document but gives also the corresponding parts within associated audio documents by exploiting the cross-linkage.

To build up a system enabling such types of application scenarios, i.e., text-to-speech synchronization, two main steps are required. In a first step, information from a given pair of documents consisting of a text document and a corresponding audio recording are analyzed. In a second step, the resulting information is then used to establish a time-alignment between the documents. To develop a reliable and efficient method, a detailed analysis of these two steps is required.

In this paper, we propose to use text-to-speech (TTS) to obtain an audio recording from a text representation, together with a subsequent audio–audio alignment based on a greedy DTW algorithm applied to feature representations obtained from the audio. The proposed method benefits from the combination of (a) suitable feature representations, (b) a greedy window-wise DTW strategy, and (c) the utilization of a TTS system for the synthe-

sizing of speech recordings. Fig. 1 shows an overview of this process. The results of the proposed method are illustrated using four different test document pairs covering different scenarios in two languages: (1) "Prolog im Himmel" from the introduction of Goethe's Faust (german language), (2) the poem "The Raven" by Edgar Allen Poe (english language), (3) a Wikipedia article about Germany (german language), and (4) a part of John F. Kennedy's famous speech "Ich bin ein Berliner" delivered on June 26, 1963 in West Berlin, Germany (english language).

The rest of this paper is organized as follows. Sect. 2 gives a detailed examination of the audio feature representation used in the context of this work. In Sect. 3, the greedy window-wise variant of DTW is introduced. Sect. 4 employs the practical evaluation and detailed analysis of our proposed synchronization method. Sect. 5 closes the paper with a conclusion and gives some prospects on future work. Related work is discussed in the respective subsection.

2. LOCAL PHRASE MATCHING AND PARAMETRIC AUDIO FEATURES

In this section, we present suitable features reflecting the phoneme progression in human speech independently of a particular speaker. These features especially meet some invariance w.r.t. speaker-individual articulation, pronunciation, and intonation.

As a standard in speech processing, mel-frequency cepstral coefficients (MFCCs) have widely been used as features in automatic speech recognition (ASR) which is the task of extracting spoken text. Furthermore, as MFCCs represent detail characteristics of individual speakers, they are also common in speaker recognition, which is the task of recognizing people from their voices. The latter property also means that MFCCs are rather sensitive to varying speakers or voices and thus less suitable for our targeted task.

In previous research, Skrowronski et al. [17] proposed human factor cepstral coefficients (HFCCs) which turned out to outperform classical MFCC features for the task of robust phoneme and speech recognition independently of the speaker. The HFCC extraction process generalizes the well-known MFCC extraction process by introducing an additional degree of freedom regarding the construction of the underlying filterbank. Whereas the filters of the classical MFCC filterbank have bandwidths determined by the center frequencies of the adjacent bands,

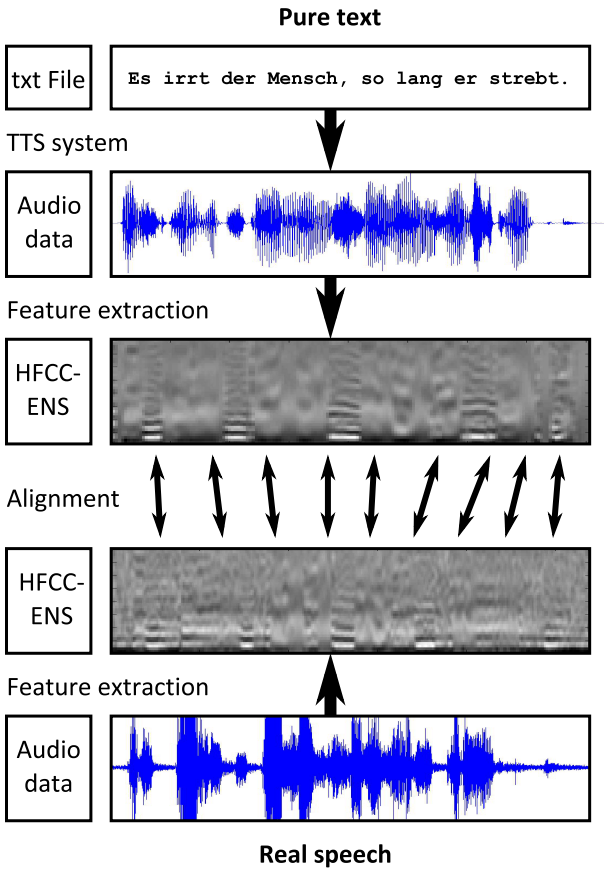

Figure 1: Schematic overview of the method used in this paper.

the authors propose to choose the bandwidth of a single filter independently of the other bands. A particular choice which was successfully applied to ASR consists of selecting the bandwidth of the mel-spaced filters according to the bark scale of human perception [17]. Fig. 2 illustrates this difference between MFCC- (top) and HFCC-based (bottom) filterbanks for the case of 16 frequency bands (the number of 16 was choosen for sake of illustration only).

In our recent research [18], we picked up the idea of controlling the used filters individually and independently from each other, and additionally postprocessed the resulting features by computing short-time energy normalized statistics (ENS). By this process, besides the spectral properties of a speech signal, also the temporal evolution of the spoken phrases is taken into account. This idea has been adopted from earlier research,

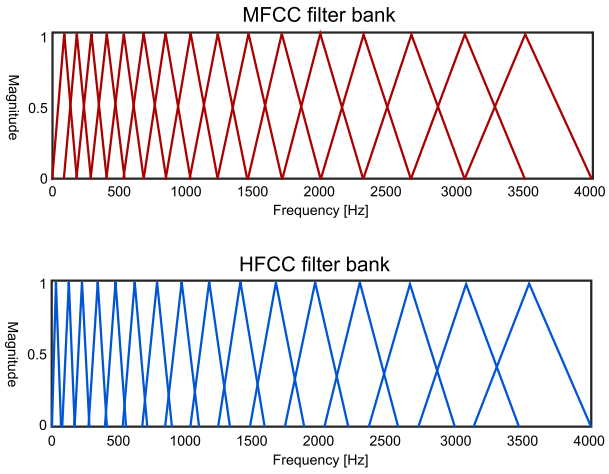

Figure 2: Comparison of MFCC- (top) and HFCC-filterbanks (bottom) with 16 bands. While the MFCCs bandwidths are determined by the centers of the neighboring filters, the HFCC-bandwidths are choosen independently.

where the inclusion of the time axis contributes a crucial level of robustness required for the identification of similar music snippets using audio matching [10]. Altogether, empirical tests showed that the resulting features are less speaker dependent and more robustly capture the phoneme progression in a sequence of spoken words. We recently employed these features successfully to unsupervised keyphrase spotting, i.e., the detection of short sequences of spoken words in a given speech signal [19].

The proposed process for extracting parametric audio features is shown in Fig. 3. Here, the front-end (frame-based spectral analysis) and back-end processing (decorrelating DCT) coincide with the well-known MFCC feature-extraction. More precisely, to compute classical mel frequency cepstral coefficients (MFCCs), an input signal is processed by a short time Fourier transform (STFT) with a block length of 20 ms and step size of 10 ms. Then, center frequencies $f_1,\ldots,f_{40}$ are choosen according to the mel scale of human pitch perception. For a fixed frame and $1 \leqslant j \leqslant J$, let $X(j)$ denote the j-th STFT-coefficient. Using triangular windows Δ_k centered at the $(f_k)_k$, spectral smoothing is performed yielding 40 mel-scale components $M(k) = \sum_{j=1}^{J} \Delta_k(j) \cdot |X(j)|$, $1 \leqslant k \leqslant 40$. To decorrelate the vector $(M(1),\ldots,M(40))$ approximately, a discrete cosine transform (DCT) is applied as a back-end processing step yielding $m = \mathrm{DCT} \cdot M$. Depending on the application, only the K-most significant coefficients $m^K = (m(1),\ldots,m(K))$ are retained for further processing (classically $K = 12$). For MFCCs, the bandwidths of the triangular filters are determined by the spacing of the center frequencies f_k, cf. Fig. 2 (top).

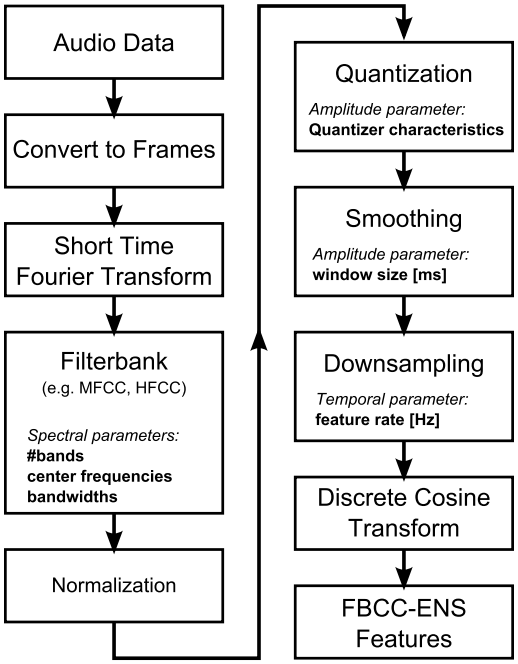

Figure 3: Detailed overview of creating features from given audio data by a filterbank approach.

To construct parametric audio features, the mel-filterbank is replaced by a general filterbank which is specified by

(i) the total frequency range,

(ii) the number of filters in this range,

(iii) the spacing of the center frequencies, and

(iv) the bandwidths of the filters.

In the special case of MFCCs, common choice are (i) a frequency range of 6500 Hz with (ii) 20–40 filters which are (iii) spaced according to the mel-scale, where (iv) the bandwidth of the i-th filter extends from the center frequency of filter $(i-1)$ to that of filter $(i+1)$. For HFCCs, only the bandwidths (iv) are exchanged and selected according to the Bark scale of critical bandwidths.

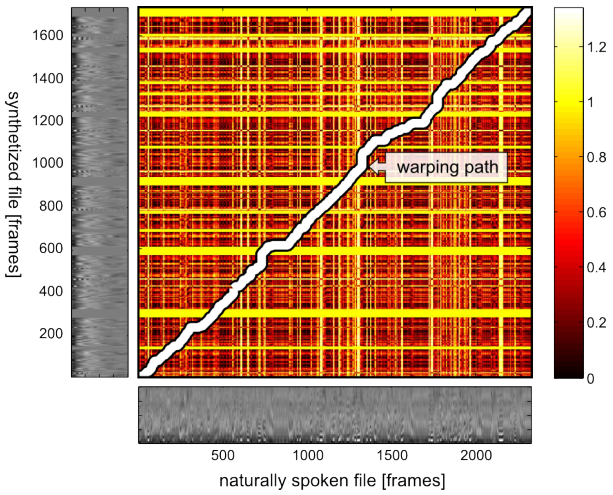

Figure 4: Classical dynamic time warping (DTW) of naturally spoken and synthetized wave files using an HFCC-ENS-based feature representation.

More precisely, the width of the bark-filter at frequency f, measured in equivalent rectangular bandwidth (ERB), is given by $E(f) = 6.23f^2 + 93.39f + 28.52$ Hz [17]. In summary, those *frequency* parameters allow us to control the spectral feature resolution.

In a subsequent process of calculating ENS, we first perform an energy normalization followed by a feature-based (component-wise) quantization of the filter bands. For normalization, the vector M is basically replaced by a normalized version $M/\sum_{k=1}^{40}|M(k)|$. The subsequent quantization in principle generalizes the log-scale compression performed in the MFCC feature-extraction. For our keyphrase-spotting application [19], we use a coarse discrete 5-step quantizer $Q : [0,1] \to \{0,1,2,3,4\}$ which is roughly logarithmic while at the same time being adapted to capture the energy-rich characteristics of phoneme transitions.

Afterwards, smoothing and downsampling of the resulting feature sequence allows us to adapt the temporal resolution of the features by choosing both the smoothing window size and the target feature resolution as *temporal* parameters. In our experiments, smoothing is performed by a Hann window. The features produced after the final DCT step will be called FBCC-ENS.

3. TEXT-TO-SPEECH ALIGNMENT

In this section, the discriminative power [19] of the feature variants described in the last section is exploited for the text-to-speech alignment task. Given a position in the text file, the task is to determine the corresponding position in the audio recording. The result can be regarded as an automated annotation of the audio recording with the given textual information.

Overall, most text-to-speech alignment techniques can be summarized in one of three different ways. A first category consists of approaches based on Hidden Markov Models (HMMs), e.g., [16]. Here, the basic idea is to model the sequence of words given by the text by creating one Markov state for each phoneme occurrence in the text and allow only one-way transitions between these states. The result is a first-order Markov model. Next, one needs observation probabilities for each state that describe the probability to observe a certain feature vector when the state is active. These probabilities are usually modelled with Gaussian mixture models (GMMs) and the associated parameters are usually estimated using supervised learning techniques. Finally, the alignment is computed using the Viterbi algorithm that determines the sequence of states that best explain the audio feature sequence observed. However, the used GMMs have to be learned from practice—oftentimes an impractible approach that further leads to very complex systems and complicates the analysis. For these reasons, we avoid an HMM-based approach in the context of our application scenario.

A second category of alignment approaches comprise methods that combine ASR techniques with text-to-text alignment methods, see for example [7]. Since ASR is much harder to solve than text-to-speech alignment, these approaches usually have to incorporate rather complex ASR systems to achieve a similar alignment accuracy as methods from the first category. Furthermore, the recognition accuracy of ASR is often insufficient. Especially in the case of low-quality audio recordings, one has to deal with a high error rate of not correctly recognized words resulting in many local mismatches thus a poor alignment. Therefore, we do not use a corresponding approach here.

In the third category one finds methods based on dynamic time warping (DTW) in combination with text-to-speech synthesis, e.g. see [13]. Here, the basic idea is to synthesize a speech recording using the given text. Then, a feature sequence derived from the original audio is compared to a sequence derived from the synthesized audio resulting in a cost matrix C. More pre-

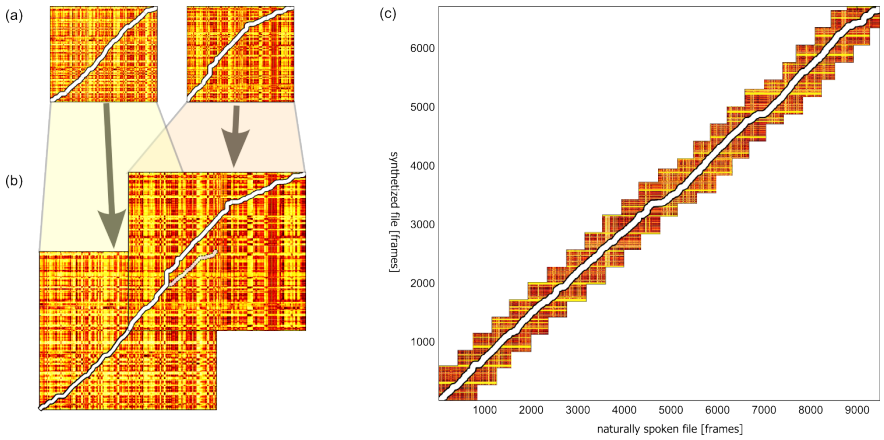

Figure 5: Window-wise computation of global DTW path by means of multiple computations of DTW sub-paths, each of which corresponding to a window. (a) shows paths for the first two windows, (b) shows a part of the overall path, constructed from sub-paths (a), and (c) elucidates the process for longer files (in this case approximately 10 minutes).

cisely, let $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$ denote the feature sequences for the original and the synthesized recording, respectively. Then, an $N \times M$ cost matrix C is built up by evaluating a local cost measure c for each pair of features, i. e., $C((n,m)) = c(x_n, y_m)$ for $n \in [1:N] := \{1, 2, \ldots, N\}$ and $m \in [1:M]$. Each tuple $p = (n,m)$ is called a *cell* of the matrix. A (global) *alignment path* is a sequence $(p_1, \ldots, p_L)$ of length L with $p_\ell \in [1:N] \times [1:M]$ for $\ell \in [1:L]$ satisfying $p_1 = (1,1)$, $p_L = (N,M)$ and $p_{\ell+1} - p_\ell \in \Sigma$ for $\ell \in [1:L-1]$. Here, $\Sigma = \{(1,0), (0,1), (1,1)\}$ denotes a set of admissible step sizes. The *cost* of a path $(p_1, \ldots, p_L)$ is defined as $\sum_{\ell=1}^{L} C(p_\ell)$. A cost-minimizing alignment path, which constitutes the final alignment result, can be computed via dynamic programming from C ([15]), see Fig. 4.

On the one hand, this third approach offers the algorithmic simplicity to concentrate on the comparison of different feature variants. On the other hand, we found it to work very well for many examples we encountered in our experiments. Hence this approach is used subsequently.

As with $N \approx M$, the memory requirements of DTW are quadratic in the length of the feature sequences. It is impossible to align very long audio recordings such as parts of Wikipedia's article about Germany with a length of over 20 minutes. To overcome this limitation, several approaches have been proposed [14, 12, 8, 9]. All trade the guarantee to find the globally optimal path off against a raise in computational efficiency.

In this paper, we follow a simple greedy strategy illustrated in Fig. 5. We start by computing C and a global alignment path $p^1 = (p_1^1, \ldots, p_{L^1}^1)$ in a small window, see Fig. 5a. Next, we move the window and set its origin to $p_{\lfloor L^1/2 \rfloor}^1$. The path $p^2 = (p_1^2, \ldots, p_{L^2}^2)$ computed in this window is used to specify the origin of the next window, see Fig. 5b. Continuing this strategy and combining all path fragments finally leads to a global path on C, see Fig. 5c. Overall, this approach does not guarantee that the globally optimal path is computed. However, using a window size of about 45 seconds, we could not find an example where the global path was different from the path computed using our window technique.

4. EVALUATION

According to the method described in the last sections and especially in Fig. 1, we tested the quality of our approach on some "real-world" data with some experimental synchronizations. Therefore we selected four text samples which exist both as a text file and as a recorded speech sound file. As feature representations for use within the synchronization algorithm we took into consideration three different feature settings, namely an MFCC-ENS-, an HFCC-ENS- and an FBCC-ENS-based feature characterization, cf. Fig 6. The feature sequences

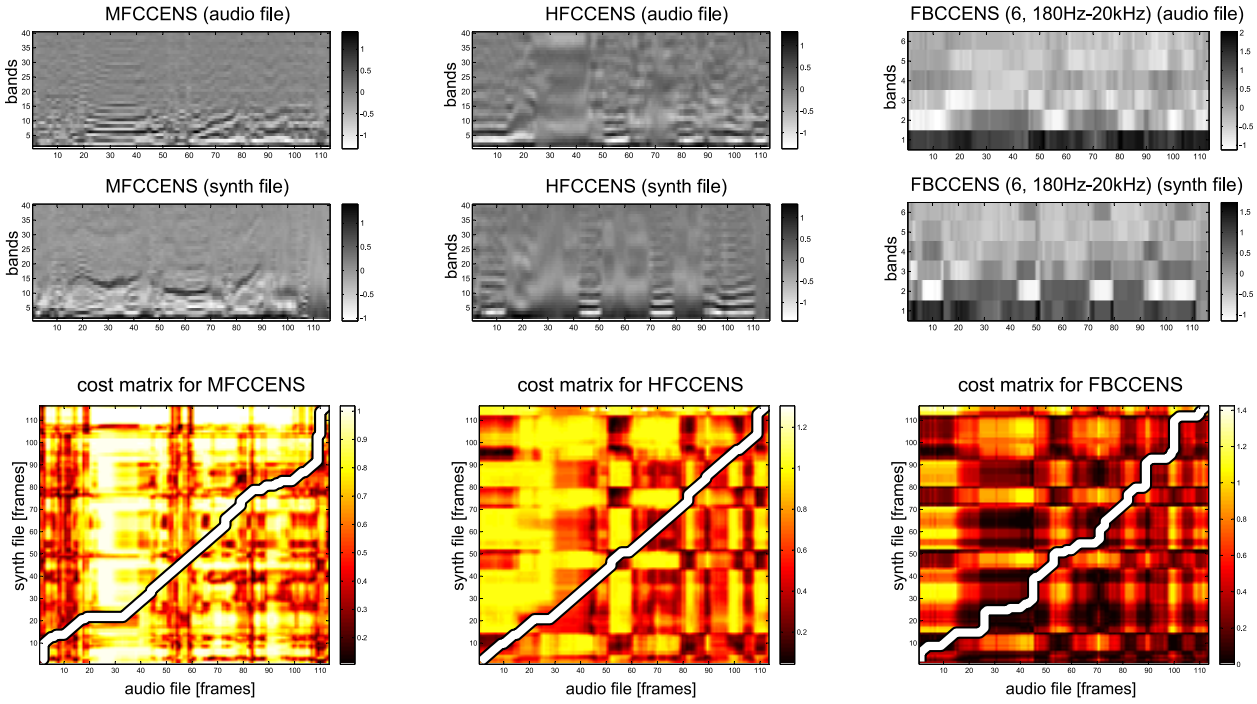

Figure 6: Comparison of (a) MFCC-ENS-, (b) HFCC-ENS-, and (c) FBCC-ENS-based feature representations (top rows), and respective local cost matrices along with their warping paths (bottom row).

in this figure are generated, respectively, from a 1.37 seconds long (real speech) and a 1.41 seconds long (synthesized speech) excerpt of text example *faust* ("Schein des Himmelslichts").

Performing these experiments consisted of the following five steps:

(i) First, the text is split up into *phrases*. A phrase is a short sentence or a subordinate clause. With regard to the importance of intonation we cannot synthesize word-for-word, but handling large sentences at a time will result in a bad time resolution. Currently, sentences as such are detected automatically, but subordinate clauses have to be taken into pieces manually.

(ii) Each phrase is processed as a whole by the TTS system. We used a commercial TTS system ("Infovox Desktop" by Acapela Group gave good results), and for simplicity we saved only starting points and durations (in milliseconds) of the phrases within the synthetized wave file. We call this information *phrase time information* and write $\mathrm{pti}(T_n)$ for the n-th phrase of text T.

(iii) A human listener created the ground truth by detecting the (nearly) exact phrase information in the wave files.

(iv) Three settings, an MFCC-ENS-, an HFCC-ENS- and an FBCC-ENS-based feature representation as shown in Fig 6, are used for synchronization. For each phrase n with phrase time information $\mathrm{pti}(T_n)$ we save the corresponding timestamps $\mathrm{pti}(T_n, F)$ for $F \in \{\text{MFCC-ENS, HFCC-ENS, FBCC-ENS}\}$.

(v) As a last step, we computed the warping path and got the corresponding starting points in the original file. These timestamps were compared to the ground truth. The *temporal aberration* of phrase n when synchronized using the features f with respect to the ground truth gT of T is defined as $\mathrm{err}(T_n, F) := \mathrm{pti}_{sp}(T_n, F) - \mathrm{pti}_{sp}(T_n, gT)$ where $\mathrm{pti}_s p$ denotes the starting point information of pti.

As text examples, we choose the following four different text pieces:

1. *faust* (german): "Prolog im Himmel" from the introduction of Goethe's Faust (audio file was downloaded from YouTube [6], text was taken from Wikisource [3]); length: 302.59 seconds, 700 words in 107 phrases,

2. *theraven* (english): The poem "The Raven" by Edgar Allen Poe (audio file [4] and text [5] were downloaded from Wikisource); length: 548.66 seconds, 1125 words in 108 phrases,

3. *wikipedia* (german): The Wikipedia article about Germany [2] spoken by a professional female reader and recorded in the context of the "WikiProject Spoken Wikipedia" [1]; length: 1214.83 seconds, 1997 words in 141 phrases,

4. *kennedy* (english): A part of John F. Kennedy's speech "Ich bin ein Berliner" delivered on June 26, 1963 in West Berlin; length: 164.83 seconds, 264 words in 30 phrases.

Within the synchronization algorithm, all features were implemented using the following *parameters* (cf. also Fig 3): winsize 60-80 ms, downsampling factor 1, and hopsize 12-15 ms. Our former experiments have shown that using those we are supposed to get the best results.

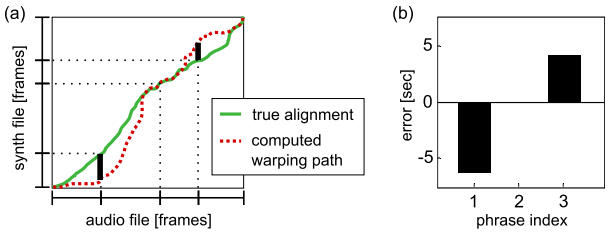

Figure 7: Coherence of (a) warping path and (b) error description $\mathrm{err}(T_n, F)$. The bars in (b) indicate the absolute temporal aberration in seconds between the real alignment (given as a ground truth by human listener) and the computed warping path.

Fig. 7 depicts an intuitive way of the illustration of the synchronization error. For each phrase n and feature setting f, the error $\mathrm{err}(T_n, F)$ is positive if the synchronized version is too fast, and negative if it is too slow. This kind of illustration will be used for our subsequent analysis. (cf. Fig. 8)

For each text piece T from the list above using F-based feature representations, the table below shows the average temporal aberration (in seconds) of synchronization $\mathbb{E}_{\mathrm{err}}(T,F) := \sum_{n=1}^{\#\mathrm{phrases}} |\mathrm{err}(T_n, F)|$.

Text	MFCC-ENS	HFCC-ENS	FBCC-ENS
faust	5.8838	0.1727	0.5778
theraven	4.5045	0.3382	0.2729
wikipedia	18.0044	0.2650	0.2121
kennedy	7.8580	9.1938	2.3885

Table 1: Mean temporal aberration $\mathbb{E}_{\mathrm{err}}(T,F)$ (in seconds) of synchronization using different feature representations

Detailed results are plotted in Fig. 8. One can observe the large errors whenever MFCC-ENS based features are used. But the greedy window-wise DTW approach is able to handle some of these: the occurring cumulative errors at *faust* and *theraven* vanishes over time. Hence we noted that even if leaving the environment of the correct warping path the synchronization may find the correct alignment in further synchronization.

Using FBCC-ENS features, we got a large error at the end of *faust*—certainly, this appeared as a delay of one single word only after a 10 seconds pause in recorded speech. Similarly, the short intervals between *theraven*'s staves caused some noticeable delays even if the HFCC-ENS-based features are used. Other errors are quite small and concerned the positions of one or two words at most.

Furthermore one observes that a synchronization of John F. Kennedy's speech was not possible by using any of the features settings above. We assume that this is a result of the totally different intonation of a public speech which is typically very tonal and exhibits many unusual word stretchings, pronunciation, and repeated words. Our approach seems to fail on this type of audio source.

The synchronization of all the other texts performed quite well using HFCC-ENS and FBCC-ENS features; not so using MFCC-ENS. Seemingly, these depend too much on the particular speaker (cf. [19]) to perform a synchronization between a real spoken text and a synthesized version of the same words. Even choosing a computer's voice of the same gender as the original text's speaker did not reduce this problem.

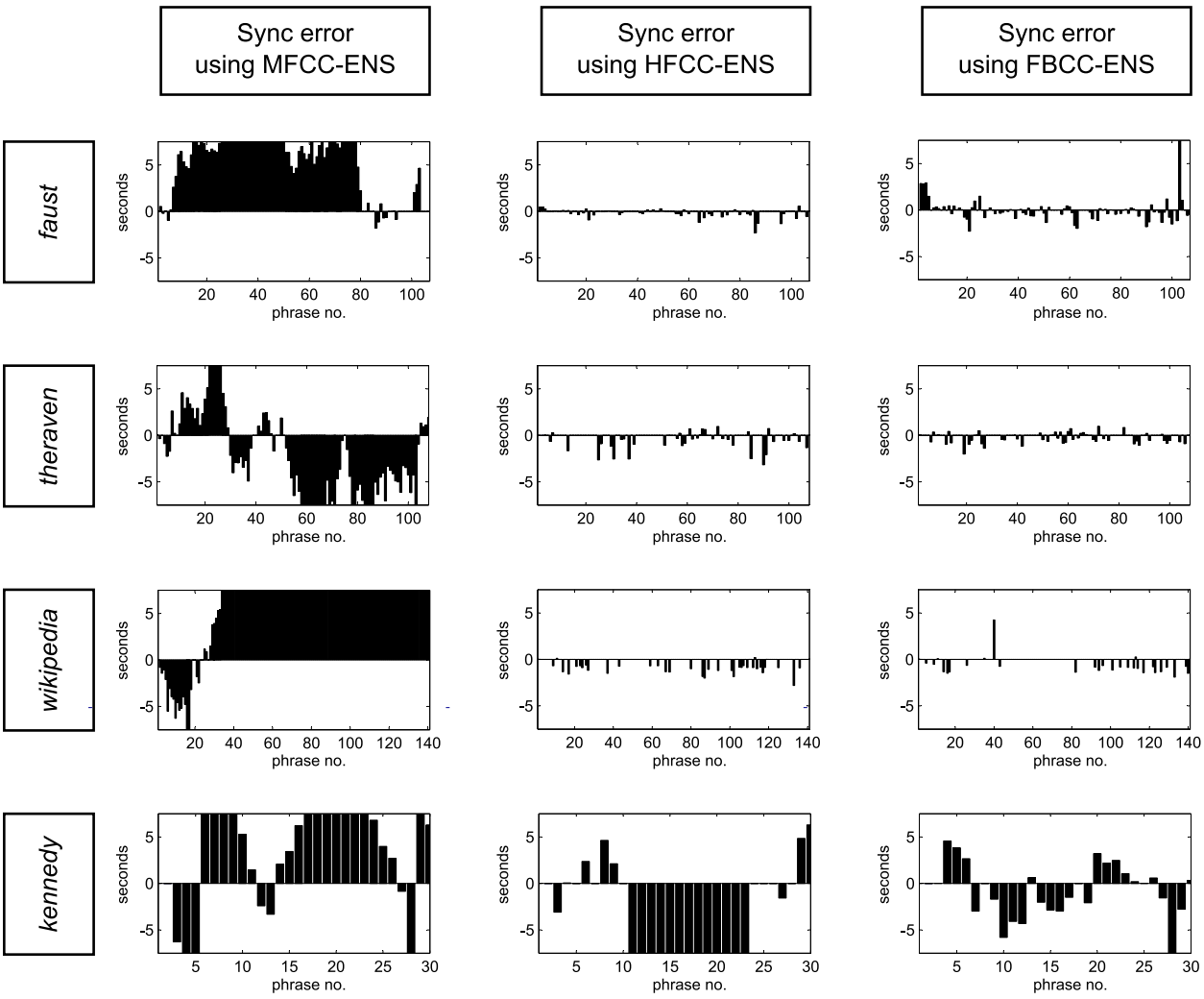

Figure 8: Overview of the temporal aberrations $\mathrm{err}(T_n, F)$ for different texts pieces T (rows) and different kinds of feature based representation F (column). For an explanation of the graphical error depiction, see Fig. 7

Using HFCC-ENS-based features obviously avoids this problem because these features outperform classical MFCC features for the taste of speech recognition independently of the speaker.

Our FBCC-ENS configuration consists of six bands only (by way of comparison, MFCC-ENS- and HFCC-ENS configurations consist of 40 bands) but occasioned surprisingly good results. Their average error is slightly smaller than those of the HFCC-ENS-based synchronization. However, as Kennedy's speech exhibits and one can observe in the graphical representation of the cost matrix in Fig 6, they are quite structurally weak compared with the many-banks supported features like MFCC-ENS or HFCC-ENS. Hence we assume that the strong drift of the computed warping path to the cost matrix' diagonal—which is typical for DTW—leads to some of the good results observed.

We selected a very short window length due to the fast changes between similar phoneme characteristics. This denies the development of macroscopic structures which may be observed in musical data. Especially, application of MFCC-ENS features with parameters as used in mu-

Figure 9: SyncPlayer with text viewer plug-in.

sic synchronization like winsize 400 ms, downsampling factor 2, and hopsize 20 ms is not possible.

For demonstration purposes, the *SyncPlayer* framework [11] was used for the time sychronous highlighting of the spoken phrase during the playback of the naturally spoken wave file, see Fig. 9. Our synchronization method described in the evaluation section created an XML file containing the phrases together with the pti information.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a method for the synchronization of a given text document and a real speech recording, both representing the same textual content. For this purpose, we follow the approach of aligning sequences of audio features derived from both input sources. Our approach combines three ingredients: First, the local comparison involved in the alignment step relies on audio feature sequences gained from both the real speech recording and a version synthesized from the text. This approach has been adopted from the area of music synchronization. Second, we pursue a greedy window-wise DTW strategy. Here, the overall alignment path is composed of partial DTW paths. Each of these are calculated on the basis of excerpts taken from the real and synthesized speech recordings, respectively. Multiple calculations of DTW paths of small snippets of the speech recordings are performed block-wise in a sliding window manner. The block-wise processing bypasses the demanding time and space complexity of a single "global" DTW on the whole speech recordings—a great drawback as speech recordings are frequently of a comparatively long time duration—and allows for the alignment of speech recordings of arbitrary lengths. Third, a TTS system is utilized to synthesize speech recordings from given text documents, where the output of the TTS system constitutes the basis for the audio feature extraction process. We used the TTS-based approach with a subsequent alignment of audio feature sequences instead of using an ASR-based approach with a subsequent alignment of text representations, because in the case of low-quality speech recordings especially, the latter approach potentially leads to a high error rate resulting in a poor alignment. In our research, we evaluated several TTS systems and discovered that the output quality of the used TTS system significantly affects feature matching quality and thus the reliability and accuracy of the alignment. The TTS system we finally used for our tests achieved significantly better results than others.

We examined how well specific feature types are suitable for the task of text synchronization. The simplicity of our DTW-based approach provides a detailed analysis of the characteristics without interference of a very complex alignment procedure such as HMM-based. Our approach does not rely on supervised learning techniques, i.e., the system need not to be trained—a big advantage as generating good training data is a time-consuming task. Various experiments showed that the alignment is quite reliable, accurate and robust as long as the text document and speech recording do not differ much in content. Since we achieved good results with our greedy window-wise DTW strategy and no training, we believe our approach is attractive for the outlined application scenario.

The poor time resolution problem and manual phrase detection discussed in Sect. 4, steps (i) and (ii), is subject to later improvement by detecting a *word time information* while using the TTS.

6. REFERENCES

[1] http://de.wikipedia.org/wiki/Datei:De-deutschland-1-article.ogg.

[2] http://de.wikipedia.org/wiki/Deutschland.

[3] http://de.wikisource.org/wiki/Faust_-_Der_Trag%C3%B6die_erster_Teil.

[4] http://en.wikisource.org/wiki/File:LibriVox_-_The_Raven_-_Chris_Goringe.ogg.

[5] http://en.wikisource.org/wiki/The_American_Review,_Volume_1,_February/The_Raven.

[6] http://www.youtube.com/watch?v=2fnPsjGJRkU.

[7] A. Haubold and J. R. Kender. Alignment of speech to highly imperfect text transcriptions. In *ICME*, pages 224–227, 2007.

[8] N. Hu, R.B. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *IEEE WASPAA (2003)*, pages 185 – 188, 2003.

[9] H. Kaprykowsky and X. Rodet. Globally optimal short-time dynamic time warping, application to score to audio alignment. In *Proceedings ICASSP 2006*, volume 5, page V, May 2006.

[10] F. Kurth and M. Müller. Efficient index-based audio matching. *IEEE TASLP*, 16(2):382–395, February 2008.

[11] F. Kurth, M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen. SyncPlayer - An Advanced System for Content-based Audio Access. In *Proceedings ISMIR, London, GB*, 2005.

[12] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings ISMIR 2007*, pages 261–266, September 2007.

[13] F. Malfrere and T. Dutoit. Speech synthesis for text-to-speech alignment and prosodic feature extraction. In *Circuits and Systems, 1997. Proceedings ISCAS '97.*, volume 4, pages 2637 –2640 vol.4, June 1997.

[14] M. Müller, H. Mattes, and F. Kurth. An Efficient Multiscale Approach to Audio Synchronization, 2006.

[15] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, United States ed. edition, April 1993.

[16] K. Sjölander. An hmm-based system for automatic segmentation and alignment of speech. In *Proceedings of Fonetik 2003*, pages 93–96, 2003.

[17] M. D. Skowronski and J. G. Harris. Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition. 116(3):1774–1780, September 2004.

[18] D. von Zeddelmann and F. Kurth. A parametric feature-based approach to noise robust speech detection for monitoring applications. In *EUSIPCO 2011*.

[19] D. von Zeddelmann, F. Kurth, and M. Müller. Perceptual audio features for unsupervised key-phrase detection. In Douglas C. Scott, editor, *IEEE ICASSP 2010 - Pt. 1*, pages 257–260, Dallas, Texas, USA, 2010.